TAMPEREEN TEKNILLINEN YLIOPISTO
TAMPERE UNIVERSITY OF TECHNOLOGY

SAMI PÄKKILÄ

MODELING AND SIMULATION OF A SIX DEGREES OF FREEDOM EXCAVATOR

Master of Science Thesis

Examiner: Professor Jouni Mattila
Examiner and topic approved in the Faculty of Engineering and Sciences council meeting on May 3rd 2017

ABSTRACT

SAMI PÄKKILÄ: Modeling and simulation of a six degrees of freedom excavator
Tampere University of technology
Master of Science Thesis, 55 pages, 10 Appendix pages
August 2017
Master's Degree Program in Automation Technology
Major: Fluid power
Examiner: Professor Jouni Mattila

Keywords: Simulation, Excavator, 6DOF, Hybrid controller, MATLAB, Simulink

In this thesis, a simulation model for a Komatsu PC138US-8 excavator was constructed. To verify the performance and validity of the model, two closed-loop control schemes were created, and they were tested with use case scenarios in which an excavator might operate at a work site. The mechanical model of the excavator was assembled in SolidWorks CAD software and imported into MATLAB Simulink environment. The hydraulics of the excavator were constructed in Simulink using the excavator's hydraulics schematic as a reference.

First off, the forward and inverse kinematics were formulated. With these kinematical equations, the excavator's operational space and joint space can be linked together, providing a foundation for the controller design implemented in this thesis. In addition to the kinematics, the dynamics of the excavator are mandatory in order to construct the desired controller schemes in this thesis. The dynamics were derived using Lagrangian equations of motion using the mass and dimension properties obtained from the CAD model.

The first use case studied in this thesis was a digging and dumping sequence. The reference trajectory was constructed from four separate sections, which were combined to form the full reference trajectory. The digging and dumping sections of the full sequence were constructed in the Cartesian space, and the bucket turning sections were constructed in the joint space. The full trajectory was then constructed in the joint space. Based on the simulation results, the excavator follows the reference trajectory well. However, fast movement on the trajectory causes the joint angles to lag the reference momentarily leading to small error in the joint value. In addition, minute steady state error is present in the joint values due to the lack of a derivative block in the implemented controller.

The second use case studied relied on the use of a hybrid position/force controller in a digging operation, where contact with the ground inflicts external force on the excavator end-effector. The x-directional position reference trajectory was given in the Cartesian space and the external contact force was given as the *z*-directional force reference. The simulation results show that the position tracking was good as the upper level position control loop kept the joint error under 0.025 radians during the digging operation. The lower level force control loop followed the force reference trajectory well, but quite large vibration was present in the controller force signal.

TIIVISTELMÄ

SAMI PÄKKILÄ: Kuuden vapausasteen kaivinkoneen mallinnus ja simulointi
Tampereen teknillinen yliopisto
Diplomityö, 55 sivua, 10 liitesivua
Elokuu 2017
Automaatiotekniikan diplomi-insinöörin tutkinto-ohjelma
Pääaine: Hydraulitekniikka
Tarkastaja: Professori Jouni Mattila

Avainsanat: Simulointi, Kaivinkone, 6DOF, Hybridisäätäjä, MATLAB, Simulink

Tässä työssä toteutettiin Komatsu PC138US-8 kaivinkoneen simulointimalli. Simulointimallin toiminnallisuuden ja oikeellisuuden varmistamiseksi kehitettiin kaksi takaisinkytkettyä säätäjäratkaisua, joita testattiin kahdella erilaisella kaivinkoneelle yleisellä käyttötapauksella. Simulointimallin mekaniikkaosio toteutettiin SolidWorks CAD ohjelmistolla, josta se tuotiin MATLAB Simulink ympäristöön. Kaivinkoneen hydrauliikka toteutettiin Simulinkissa kaivinkoneen olemassaolevaa hydraulikaaviota hyväksikäyttäen.

Ensiksi, kaivinkoneen suoran ja käänteisen kinematiikan yhtälöt johdettiin, joiden avulla kaivinkoneen karteesinen avaruus ja nivelavaruus voidaan linkittää keskenään. Tällä yhteydellä saadaan luotua pohja tässä työssä kehitettyjen säätäjäjien toteutukseen. Säätäjien toteuttamista varten kinematiikkojen lisäksi kaivinkoneen dynamiikan tuntemus on välttämätöntä. Tässä työssä kaivinkoneen dynamiikka johdettiin käyttäen Lagrange'n liikeyhtälöitä sekä massa ja dimensio tietoja olemassa olevasta CAD mallista.

Työn ensimmäinen käyttötapaus oli kaivinkoneen kaivuu sekä kuorman kippaus työsykli. Työsyklin referenssi rata koostui neljästä yksittäisestä osasta, jotka lopulta yhdistettiin yhdeksi liikeradaksi. Liikeradan kaivuu sekä kuorman kippaus osuudet toteutettiin karteesisessa avaruudessa ja kauhan kääntö osuudet nivelavaruudessa. Lopullinen koko liikerata toteutettiin nivelavaruudessa. Simulointitulosten perusteella kaivinkoneen simulaatiomalli toteuttaa referenssi liikerataa hyvin. Siitä huolimatta, nopeat liikkeet aiheuttavat hetkellistä viivettä ja siten virhettä nivelkulman arvoihin. Lisäksi, nivelissä esiintyy hyvin pientä virhettä tasapainotilassa johtuen derivaatiolohkon puutteesta toteutetussa säätäjäratkaisussa.

Toinen tutkimuksen kohteena ollut käyttötapaus perustui hybridi asema/voimasäätäjän käyttöön kaivuuoperaatiossa, jossa maan vetovastus ja kauhan täyttyminen aiheuttavat ulkoista voimaa kaivinkoneen kauhaan. Kaivinkoneen x-suuntainen asemareferenssi annettiin karteesisessa tasossa ja z-suuntaisena voimareferenssi käytettiin kaivinkoneen kauhaan aiheutuvaa ulkoista voimaa. Käyttötapauksen simulaatiotulokset todentavat, että asemasäätäjän referenssin seuranta oli hyvää nivelten maksimivirheen jäädessä 0.025 radiaaniin. Alemman tason voimasäätäjä seurasi myös referenssiään hyvin, mutta suhteellisen suurta värähtelyä esiintyi säätäjän voimasignaalissa.

PREFACE

This Master of Science Thesis was undertaken at Tampere University of Technology (TUT) at the laboratory of Automation and Hydraulic Engineering (AUT) in the spring and summer of 2017.

I would like thank Prof. Jouni Mattila for this great opportunity to work at AUT and for his guidance and mentoring during this thesis. I also want to thank my colleagues for creating a pleasant working environment and for their relentless support throughout this journey.

I am very grateful for the support I have received from friends and family over the years and would like to extend my deepest gratitude. I especially want to thank my parents for their constant support in academic life and outside of it.

Tampere, 23.8.2017

Sami Päkkilä

CONTENTS

NOMENCLATURE AND ABBREVIATIONS

CAD	Computer-aided Design
DH	Denavit-Hartenberg notation for kinematic parametrization
DOF	Degrees of freedom
MATLAB	Multi-paradigm numerical computing environment and programming language created by MathWorks
PI	Proportional-integral
Pose	End-effector position and orientation
TUT	Tampere University of Technology

$\mathbf{A}_i^{i-1}$	Transformation matrix from frame $\mathbf{A}_{i-1}$ to $\mathbf{A}_i$
a_i	DH-parameter length of link i
α_i	DH-parameter angle between axes z_i and z_{i-1} about axis x_i
C	Vector of Coriolis and centrifugal terms
{C}	Constraint frame in Cartesian space
CoG	Link center of gravity
D	Inertia matrix
d_i	DH-parameter distance between two links measured along axis z_i
ε	Unit quaternion vector part
$\mathbf{F}_f$	Actuator friction force
ϕ	Euler Roll-Pitch-Yaw angles $\phi = [\varphi\ \upsilon\ \psi]^T$
$\mathbf{F}_r$	External reaction force
G	Gravity torque vector
I	Identity matrix
J(θ)	Jacobian matrix
K	Kinetic energy
L	Lagrange equation of motion
η	Unit quaternion scalar part
p	Position vector $\mathbf{p} = [p_x\ p_y\ p_z]^T$
P	Potential energy
Q	Unit quaternion angle $Q = \{\eta,\varepsilon\}$
$\mathbf{q}_i$	Joint variable
Q	Controller command value vector
R	Rotation matrix
S	Compliance selectivity matrix
T	Transformation matrix
θ_i	DH-parameter joint variable for revolute joint
$\theta_d(t)$	Desired joint value
$\theta(t)$	Actual joint value
τ_i	Joint torque

1. INTRODUCTION

Excavators have been the backbone of urban construction, agriculture, forestry and mining industries for decades. They come in a wide range of sizes starting from mini excavators weighing in at a ton up to the larger mining equipment at almost 980 tons [1]. Excavators are comprised of two main sections: the undercarriage and the swing. The undercarriage includes the tracks, track frame and final drives, which provide drive to the individual tracks. On occasion, the undercarriage may also be fitted with a blade attached on the front end. The swing section includes the operator cabin, a counterweight, a diesel engine as well as fuel and hydraulic oil tanks. The undercarriage and the swing are connected via a center pin allowing the swing to slew about the undercarriage unconstrained.

Connected to the swing is the main boom, which typically is limited to operate only in the vertical direction other configurations being available also. Attached to the end of the main boom is the arm, which provides the necessary digging force to pull the bucket through the ground. Depending on the individual needs, the length of the arm and main boom can be chosen from a set of options. Shorter dimensions provide more stability and breakout force while longer options give extended reach with the cost of smaller bucket sizes.

Generally, excavators are equipped with a bucket attached to the end of the arm, but several other options, such as grapplers, breakers and saws, are available depending on the task at hand. Furthermore, many excavators are fitted with a quick coupler allowing easy tool changes on site making them highly versatile on the jobsite. Due to the wide availability of tools, excavators are used in a wide range of tasks including digging, landscaping, construction, material handling, drilling etc.

In modern day society, the role of excavators has only increased with the introduction of more versatile tools and intelligent control systems to further assist the operator in daily tasks at the site. Where the operator previously had to rely solely on visual feedback from the current task, nowadays the increased use of sensors combined with a computer system allow the operator to monitor the position and orientation of the tool resulting in improved precision and efficiency. Furthermore, without any measured feedback the operator is required be more skilled in operating the excavator than his aid-using counterpart and the working hours are limited due to visibility. Not being confined to monitoring only, with modern computer based control systems different control methods, such as Cartesian position control, are available for the operator to choose from.

1.1 Komatsu PC138US-8 excavator

The excavator studied in this thesis is a 13.5-ton Komatsu PC138US-8 model (figure 1) equipped with a 0.5 m^3 backhoe bucket and a mRoto-10 tiltrotator manufactured by Marttiini Metals. The excavator consist of the above mentioned two main sections, the undercarriage and the swing, joined by the main boom, arm, tiltrotator and bucket. The undercarriage is driven by two hydraulic piston type motors and has a hydraulic track adjuster and parking brake. The width of the crawler (undercarriage) is 2490 mm and it has a track length of 3610 mm. The steering of the excavator is fully hydrostatic with two levers with pedals. Top speed of 2.9 km/h is achieved on the low gear and 5.1 km/h on the high gear.

The swing section contains a water-cooled, turbocharged diesel engine outputting 68.4 kW of power driving all the hydraulic circuits. The hydraulic pumps, the fuel and oil tanks as well as the counterweight are located behind the operator cabin. In unison with modern excavator design, the swing section is relatively short in length giving the excavator a small tail swing radius of 1480 mm, which is especially convenient when operating in tight or confined spaces. A hydraulic motor, equipped with a swing lock rotates the swing section unconstrained.

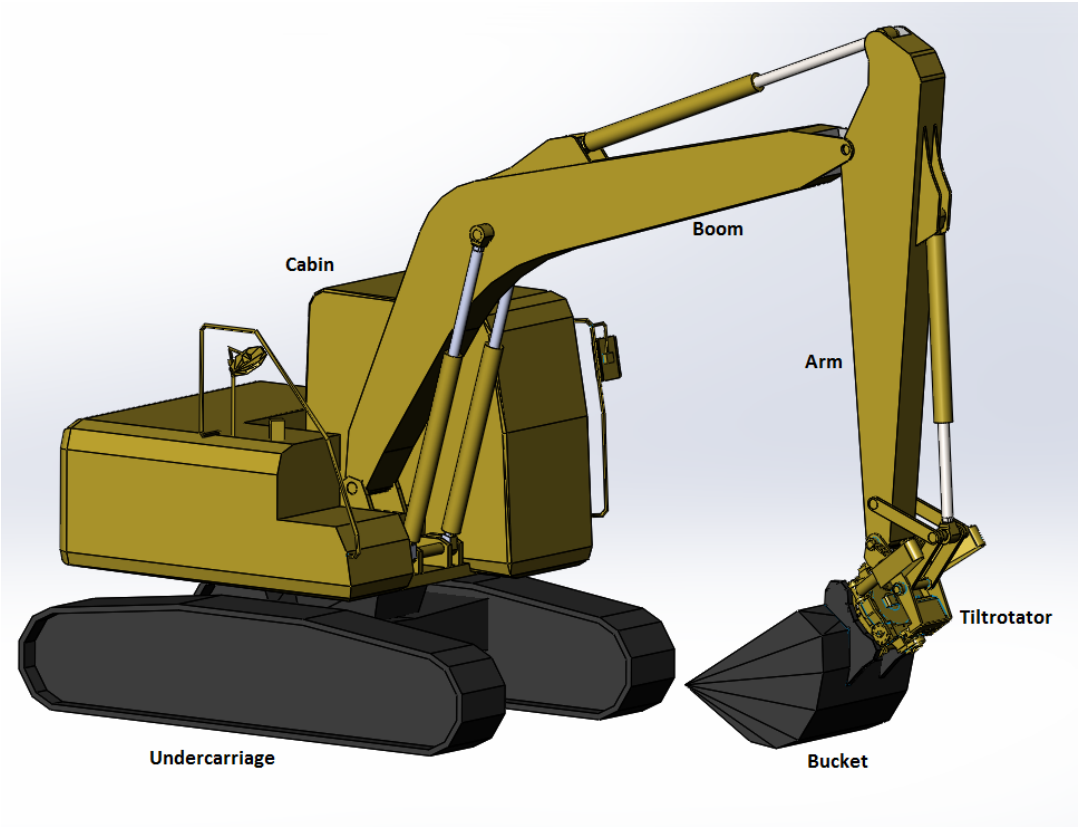

Figure 1. Excavator overview

The main boom is 4.6 meters in length and is operated by two dual acting hydraulic cylinders. Connected to the boom is the 2.5 meter long arm and at the end the tiltrotator. The tiltrotator is connected to the arm by a four-bar linkage and both the arm and tiltrotator are operated by dual acting hydraulic cylinders. The tiltrotator can pivot to both sides up to 40 degrees and it can rotate unconstrained moving the bucket attached to it simultaneously. [16]

1.2 Computer based modeling

Building a complete measurement and control system to an actual machine can be very time consuming and expensive. In addition to all the required sensors and onboard electronics, one must also procure the machine itself. With modern computer software, building a simulation model of the actual machine is relatively straightforward and comes with several benefits.

The main focus when building a simulated environment is to approximate the actual machine to a certain extent. If the physical dimensions and masses of the actual machine are known, construction of the mechanical model will result in a relatively accurate simulation model. However, depending on the requirements and the final use circumstances, approximate dimensions and masses might still lead to a sufficiently accurate result.

The major advantage of using simulation models is their high degree of modification. Parameter changes and modifications to the model itself are easy to make and lead to instant results. Troubleshooting for issues is simple as any signal, for example, pressure, force, velocity or oil flow, can be plotted from virtually any given location in the model. In addition, the entirety of the model can be broken down into subsections or some features, for example, actuators or machine links can be opted out of the simulation model.

An important feature related to simulation models is repeatability. Simulations run with one set of inputs will give the same results, since no external disturbance affects the simulation. This makes tuning of the parameters easier and enables reliable comparison between different simulation runs. Another great advantage with computer simulation models compared to working with actual machines is the safety aspect. Even if some parameters are far off, no harm will be done neither to the machine nor to the operator.

Even though simulation models are an excellent tool to create fairly accurate representations of the actual machine, they have some elementary drawback to them. No matter how precisely the model is constructed and how accurate the parameters are, it still is only a simulated model. Simulation models are an idealization of the actual machine and the environment it is in and for example, prevalent weather conditions are not taken into account in any way. In addition, some phenomena are very difficult to accurately model, for example friction, the effects of temperature to fluid viscose and fluid dynamics in general. Building simulation models also requires some level of expertise, especially if

the goal is to get results that can be transformed later on to the actual machine environment.

1.3 Goals for the thesis

The first and foremost goal of this thesis is to create a simulation model consisting of both the mechanical structure and the hydraulic circuitry in the MATLAB Simulink environment based on the Komatsu PC138US-8 excavator and implement appropriate control methods to operate the excavator in specific use case situations. The second goal of this thesis, which is directly related to the first goal, is to create a kinematic representation of the excavator linking the operational space and joint space together. In addition, the equations of motion, namely the excavator dynamics, need to be formulated to control the excavator during movement.

Creating the different controller configurations and running simulations implementing them, which is the third goal of this thesis, will give insight on the behavior of the excavator and provide information about the modeling validity. The fourth goal of this thesis is to create a foundation to continue the computer based simulation of the excavator and eventually extend it to including implementation of the controller schemes to the real Komatsu excavator.

1.4 Structure of the thesis

This thesis consist of six chapters. After the introduction, in chapter two, the forward and inverse kinematics for the excavator are formulated. Then, in chapter three, the foundation of manipulator equations of motion is discussed and the excavator dynamics equations are formulated. These dynamics equations are implemented in chapter four in the construction of the controller design. In addition, in chapter four, path planning in both the Cartesian space and the joint space is presented. In chapter five, simulations of the model constructed in this thesis are run utilizing the different controller schemes and trajectories created in chapter four. Lastly, conclusions concerning the results and observations regarding the thesis are presented in chapter six. Moreover, future work and possible real life implementation are discussed in this chapter.

2. KINEMATICS

A manipulator can be described as rigid body links in series or in parallel configuration connected to each other by either revolute or prismatic joints. These consecutive links form a chain, which is constrained to a fixed base frame in one end and on the other end of the chain is an end effector, for example a gripper or a tool, which allows manipulation of objects in space. A manipulator's kinematic chain can be either open, in which case the body links form a chain that has two ends, or it can be closed, in which case the body links form a loop. [2]

The manipulator's mechanical structure is determined by its degrees of freedom, which typically is associated to the number of joints in the kinematic chain. In order to move the manipulator in space, its end effector's position and orientation must be expressed as a function of individual joint variables with respect to a reference coordinate frame. This can be done with the forward kinematics equation, which is discussed in section 2.1. It is also important to be able to express the given position and orientation of the manipulator as joint variables. This is called the inverse kinematics problem and it is discussed in section 2.2.

The forward and inverse kinematics do not directly take into account the effects of the environment, such as gravitational force, on the manipulator. Moreover, for example in the case of an excavator, the contact force between the bucket and the ground in a digging motion affects the torque directed on the joints, which has to be compensated. This problem is related to the dynamics of the manipulator and it will be discussed in section 2.3.

2.1 Forward kinematics

Manipulator forward kinematics express the position and orientation of the end effector in terms of individual joint variables. Each link in the kinematic chain has to be assigned a unique coordinate frame, which allow expressing each link in terms of rotation and translation with respect to a reference frame. The position and orientation of the end effector can be determined with these individual coordinate frames by means of matrix multiplication.

2.1.1 Position vector and rotation matrix

In order to express the position and orientation of an object in space, a coordinate frame has to be assigned to it. This requires the position and orientation to be known with respect to a reference frame. The manipulator position in space can be expressed with a position vector **p** from the reference frame to the object's coordinate frame.

$$\boldsymbol{p} = \begin{bmatrix} p_x & p_y & p_z \end{bmatrix}^T \tag{2.1}$$

The orientation of the object with respect to the reference frame can be expressed with a rotation matrix **R** (2.5), which is composed of elementary rotations about the *x*-*y*- and *z*-axes. These rotations are positive if they are made counter-clockwise about the relative axis. In figure 2, coordinate frame *O-xyz* is rotated by angle α about axis *z* making the coordinate frame *O-x'y'z'* the rotated frame.

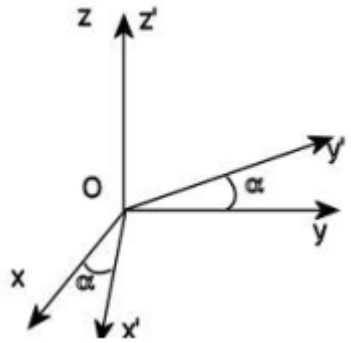

Figure 2. Rotation of coordinate frame O-xyz about the z-axis

The unit vectors of the new frame with respect to the reference frame can be described as:

$$\boldsymbol{x'} = \begin{bmatrix} cos(\alpha) \\ sin(\alpha) \\ 0 \end{bmatrix}, \quad \boldsymbol{y'} = \begin{bmatrix} -sin(\alpha) \\ cos\,(\alpha) \\ 0 \end{bmatrix}, \quad \boldsymbol{z'} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

Hence, the rotation matrix of frame *O-x'y'z'* with respect to frame *O-xyz* is

$$\boldsymbol{R}_z(\alpha) = \begin{bmatrix} \cos(\alpha) & -\sin(\alpha) & 0 \\ \sin(\alpha) & \cos(\alpha) & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{2.2}$$

Rotations by an angle β about axis *y* and by angle γ about axis *x* can be expressed in a similar manner as:

$$\boldsymbol{R}_y(\beta) = \begin{bmatrix} \cos(\beta) & 0 & \sin(\beta) \\ 0 & 1 & 0 \\ -\sin(\beta) & 0 & \cos(\beta) \end{bmatrix} \tag{2.3}$$

$$\boldsymbol{R}_x(\gamma) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\gamma) & -\sin(\gamma) \\ 0 & \sin(\gamma) & \cos(\gamma) \end{bmatrix} \tag{2.4}$$

Multiplying the above defined elementary rotation matrices yields the complete rotation matrix.

$$\boldsymbol{R} = \boldsymbol{R}_x(\gamma)\boldsymbol{R}_y(\beta)\boldsymbol{R}_z(\alpha) \tag{2.5}$$

The matrix **R** describes the rotation about an axis in space needed to align the axes of the reference frame with the corresponding axes of the body frame [2]. An important feature of the rotation matrix **R** is that it is orthogonal. This means that

$$\boldsymbol{R}^T\boldsymbol{R} = \boldsymbol{I} \tag{2.6}$$

where **I** denotes the *(3x3)* identity matrix.

Postmultiplying both sides of equation (2.6) leads to a useful result

$$\boldsymbol{R}^T = \boldsymbol{R}^{-1} \tag{2.7}$$

Combining the position vector and the rotation matrix produces a *(4x4)* transformation matrix.

$$\boldsymbol{A}_i^{i-1} = \begin{bmatrix} \boldsymbol{R}_i^{i-1} & \boldsymbol{p}_i^{i-1} \\ 0^T & 1 \end{bmatrix} \tag{2.8}$$

This transformation matrix **A** expresses the transformation of the position and orientation when moving from coordinate frame $\mathbf{A}_{i\text{-}1}$ to $\mathbf{A}_i$.

2.1.2 Denavit-Hartenberg Convention

The Denavit-Hartenberg convention (figure 3) is a widely used four parameters presentation that allows the computation of the relative position and orientation between two consecutive links of a manipulator. It utilizes the unique coordinate frames attached to each link in the kinematic chain, which in general can be arbitrarily chosen as long as they are attached to the link they are referred to. However, it is convenient to set some rules when defining the link coordinate frames associated with the Denavit-Hartenberg convention, for example, letting the joint variables always be constrained to the relative *z* axis.

The first DH-parameters is a_i, which is the length of link *i*. It is measured by the distance along the common normal of axes z_i and $z_{i\text{-}1}$. The second DH-parameter is α_i, which is the angle between the axes $z_{i\text{-}1}$ and z_i about the axis x_i. The angle α_i is positive when rotation is made counter-clockwise. The third parameter is d_i, which is the distance between two links measured along axis $z_{i\text{-}1}$. Finally, the fourth parameter is angle θ_i, which is the angle between axes $x_{i\text{-}1}$ and x_i about the axis $z_{i\text{-}1}$. As with the angle α_i, θ_i is taken positive when rotation is made counter-clockwise.

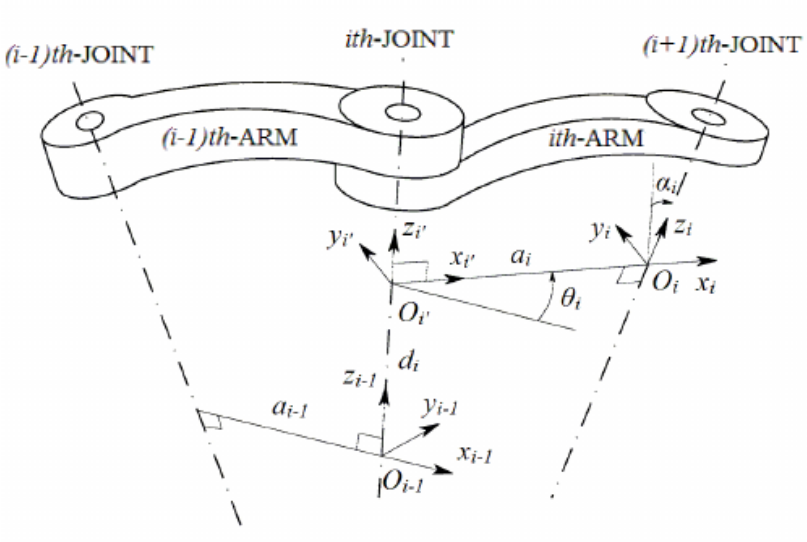

Figure 3. Parameters for a revolute joint in the Denavit-Hartenberg convention [3]

The joint variables in the Denavit-Hartenberg convention are θ_i in the case of a revolute joint and d_i in the case of a prismatic joint. The remaining three parameters are considered as constants. The excavator coordinate frames associated with the DH-parameters are shown in figure 4.

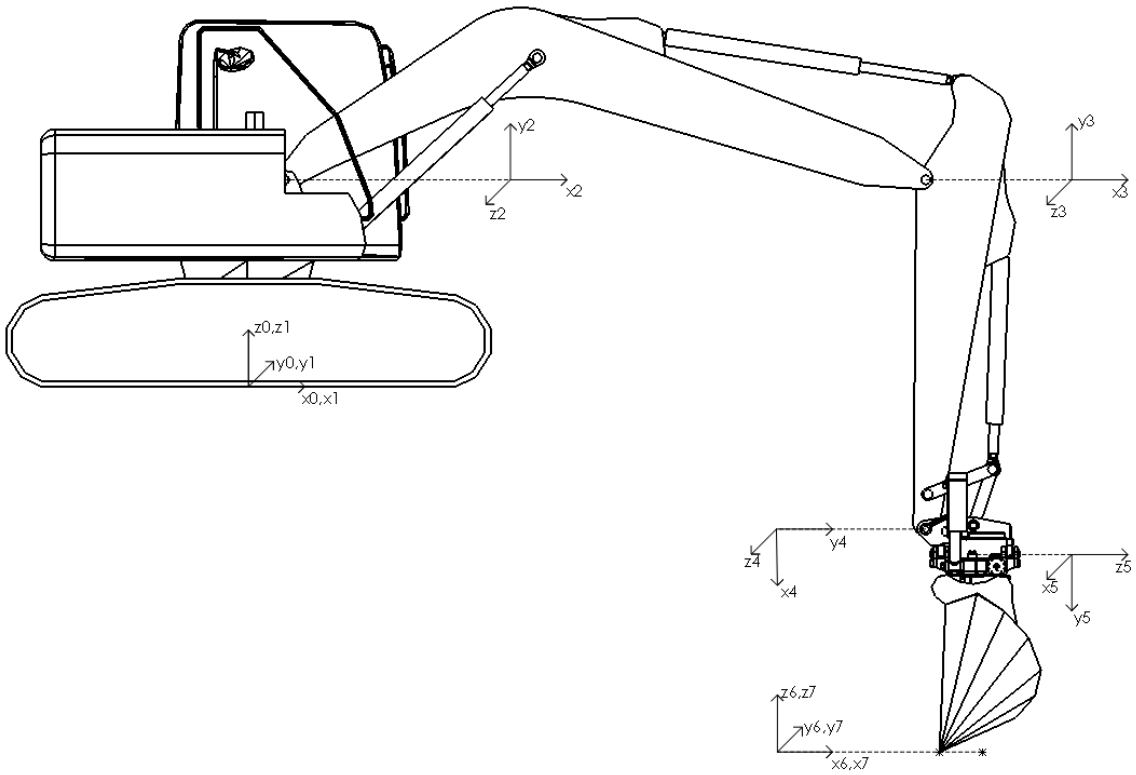

Figure 4. Excavator coordinate frames

The excavator is comprised of six consecutive revolute joints, which makes the joint variable for each coordinate frame θ_i. The excavator's DH-parameters are shown in table 1.

Table 1. Denavit-Hartenberg parameters for excavator

Joint	α_i	a_i	d_i	θ_i
1	$\pi/2$	a_1	d_1	θ_1
2	0	a_2	d_2	θ_2
3	0	a_3	0	θ_3
4	$-\pi/2$	a_4	0	θ_4
5	$\pi/2$	0	d_5	$\theta_5 - \pi/2$
6	0	0	$-d_6$	$\theta_6 + \pi/2$
7	0	$-a_7$	0	0

Joints one through six are the six revolute joints of the excavator and joint seven is the tool transfer frame to the front edge of the bucket. The end effector frame has conveniently been aligned with the base frame.

The coordinate transformation between frames *i-1* and *i* can now be expressed in two steps. First frame *i-1* is translated by d_i along axis z_{i-1} and rotated by θ_i about axis z_{i-1}. This aligns the current frame with frame *i'* and it can be described by the homogenous transformation matrix

$$\mathbf{A}_{i'}^{i-1} = \begin{bmatrix} \cos(\theta_i) & -\sin(\theta_i) & 0 & 0 \\ \sin(\theta_i) & \cos(\theta_i) & 0 & 0 \\ 0 & 0 & 1 & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

The frame aligned with frame *i'* is then translated by a_i along axis x_i and rotated by α_i about axis x_i. This aligns the current frame with frame *i* and it can be described by the homogenous transformation matrix

$$\mathbf{A}_{i}^{i'} = \begin{bmatrix} 1 & 0 & 0 & a_i \\ 0 & \cos(\alpha_i) & -\sin(\alpha_i) & 0 \\ 0 & \sin(\alpha_i) & \cos(\alpha_i) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

By multiplying the single transformations, the overall coordinate transformation from frame *i-1*to *i* is obtained as

$$\mathbf{A}_i^{i-1}(q_i) = \mathbf{A}_{i''}^{i-1}\mathbf{A}_i^{i''} = \begin{bmatrix} \cos(\theta_i) & -\sin(\theta_i)\cos(\alpha_i) & \sin(\theta_i)\sin(\alpha_i) & a_i\cos(\theta_i) \\ \sin(\theta_i) & \cos(\theta_i)\cos(\alpha_i) & -\cos(\theta_i)\sin(\alpha_i) & a_i\sin(\theta_i) \\ 0 & \sin(\alpha_i) & \cos(\alpha_i) & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2.9}$$

In addition to the homogenous transformation matrix, its inverse is needed in later calculations. The inverse can be described as

$$inv\mathbf{A}_i^{i-1} = \left[\mathbf{A}_i^{i-1}\right]^{-1} = \begin{bmatrix} \cos(\theta_i) & \sin(\theta_i) & 0 & -a_i \\ -\sin(\theta_i)\cos(\alpha_i) & \cos(\theta_i)\cos(\alpha_i) & \sin(\alpha_i) & -d_i\sin(\alpha_i) \\ \sin(\theta_i)\sin(\alpha_i) & -\cos(\theta_i)\sin(\alpha_i) & \cos(\alpha_i) & d_i\cos(\alpha_i) \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2.10}$$

The transformation matrix **T** from the base frame to the end effector frame can be obtained by multiplication of all the individual homogenous transformation matrices as

$$\mathbf{T}_7^0(\mathbf{q}) = \mathbf{A}_1^0\mathbf{A}_2^1\mathbf{A}_3^2\mathbf{A}_4^3\mathbf{A}_5^4\mathbf{A}_6^5\mathbf{A}_7^6 = \begin{bmatrix} \mathbf{n}^0 & \mathbf{s}^0 & \mathbf{a}^0 & \mathbf{p}^0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \tag{2.11}$$

where **q** is the joint variable, and

$$\mathbf{p}^0 = \begin{bmatrix} c_1(s_5s_6a_7 + c_5d_6)c_{234} + c_1(c_6a_7 - d_5)s_{234} + a_4c_{23}c_1c_4 - a_4s_{23}c_1s_4 + ((c_3a_3 + a_2)c_2 - s_2s_3a_3 + a_1)c_1 + s_1(c_5s_6a_7 - s_5d_6 + d_2) \\ s_1(s_5s_6a_7 + c_5d_6)c_{234} + s_1(c_6a_7 - d_5)s_{234} + a_4c_{23}c_4s_1 - a_4s_{23}s_1s_4 + ((c_3a_3 + a_2)c_2 - s_2s_3a_3 + a_1)s_1 - c_1(c_5s_6a_7 - s_5d_6 + d_2) \\ d_1 - \frac{1}{2}a_7c_{2346} + \frac{1}{2}d_6s_{2345} + a_2s_2 - \frac{1}{2}a_7c_{234-6} + \frac{1}{2}d_6s_{234-5} + d_5c_{234} + a_4c_{23}s_4 + a_4s_{23}c_4 + a_3c_2s_3 + a_3c_3s_2 + a_7s_{234}s_5s_6 \end{bmatrix}$$

the end-effector position, and

$$\mathbf{n}^0 = \begin{bmatrix} -s_6(c_5s_1 + c_{234}c_1s_5) - s_{234}c_1c_6 \\ s_6(c_1c_5 - c_{234}s_1s_5) - s_{234}c_6s_1 \\ \frac{1}{2}c_{2346} + \frac{1}{2}c_{234-6} - s_{234}s_5s_6 \end{bmatrix}$$

$$\mathbf{s}^0 = \begin{bmatrix} s_{234}c_1s_6 - c_6(c_5s_1 + c_{234}c_1s_5) \\ c_6(c_1c_5 - c_{234}s_1s_5) + s_{234}s_1s_6 \\ -\frac{1}{2}s_{2346} + \frac{1}{2}s_{234-6} - s_{234}c_6s_5 \end{bmatrix}$$

$$\mathbf{a}^0 = \begin{bmatrix} s_1s_5 - c_{234}c_1c_5 \\ -c_1s_5 - c_{234}c_5s_1 \\ -\frac{1}{2}s_{2345} - \frac{1}{2}s_{234-5} \end{bmatrix}$$

the orientation, where $c_1 = \cos(\theta_1)$, $s_1 = \sin(\theta_1)$, $c_{234} = \cos(\theta_2 + \theta_3 + \theta_4)$, $s_{234\text{-}6} = \sin(\theta_2 + \theta_3 + \theta_4 - \theta_6)$ and so forth.

2.2 Inverse kinematics

The forward kinematics map the relationship between the manipulator joint variables and the end-effector position and orientation. The inverse kinematics on the other hand attempts to determine the joint variables corresponding to a given end-effector position and orientation. Controlling of manipulators in generally executed in the Cartesian operational space, making it vitally important to be able to convert the motion specifications from the operational space into the corresponding joint space motions enabling the desired motions to be executed.

While the forward kinematics equation produces a unique solution for the end-effector position and orientation provided that the joint variables are known, solution for the inverse kinematics is much more complicated. For starters, the equations to solve are in general nonlinear; ergo it is not always possible to find a closed-form solution. Moreover, multiple solutions may exist or even infinite solution if the manipulator is kinematically redundant. Depending on the manipulator kinematic structure, it is also possible that no solutions exist, which is usually the case when the end-effector position and orientation are outside the manipulators admissible workspace. If no constraints are applied to the joint variables, a six degrees of freedom manipulator has in general up to 16 admissible solutions for the inverse kinematics problem. However, the kinematics constraints of the actual machine reduce the number of multiple solutions. [2]

2.2.1 Solution of six degrees of freedom excavator

The inverse kinematics for the six degrees of freedom excavator were acquired by means of algebraic methods. Given a certain position and orientation desired for the end-effector in the base coordinate frame, the pose can be obtained by making the joint variables assume the values that result in it. This can be achieved by substituting the coordinates of the end-effector in the base coordinate frame into the kinematic equations defined in equation (2.11) [4]. Rewriting equation (2.11) yields

$$\mathbf{T}_7^0 = \begin{bmatrix} n_x & s_x & a_x & p_x \\ n_y & s_y & a_y & p_y \\ n_z & s_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} = \mathbf{A}_1^0 \mathbf{A}_2^1 \mathbf{A}_3^2 \mathbf{A}_4^3 \mathbf{A}_5^4 \mathbf{A}_6^5 \mathbf{A}_7^6, \tag{2.12}$$

where the terms on the right-hand side of the equation are in terms of the structural kinematic parameters and joint variables and the terms on the left-hand side are known parameters related to the case under consideration.

Solving the joint variables θ_i requires the equation (2.12) to be manipulated in such a manner that the unknown variable will be isolated on the left hand side of the equation. This can be done by premultiplying both sides of the equation with the inverse of the matrix appearing as the first term in the product on the right side of the equation.

Solving for θ_1, both sides of equation (2.12) are premultiplied by $\left[\mathbf{A}_1^0\right]^{-1}$ leading to

$$\left[\mathbf{A}_1^0\right]^{-1}\mathbf{T}_7^0 = \mathbf{A}_2^1\mathbf{A}_3^2\mathbf{A}_4^3\mathbf{A}_5^4\mathbf{A}_6^5\mathbf{A}_7^6 \tag{2.13}$$

Setting the elements [3,4] of both sides equal yields

$$\mathbf{p}_x \sin(\theta_1) - \mathbf{p}_y \cos(\theta_1) = d_2 - d_6 \sin(\theta_5) + a_7 \cos(\theta_5)\sin(\theta_6)\,. \tag{2.14}$$

Substituting elements [3,3] and [3,1] into equation (2.14) leads to

$$\mathbf{p}_x \sin(\theta_1) - \mathbf{p}_y \cos(\theta_1) = d_2 - d_6(\mathbf{a}_x \sin(\theta_1) - \mathbf{a}_y \cos(\theta_1)) + a_7(\mathbf{n}_y \cos(\theta_1) - \mathbf{n}_x \sin(\theta_1))\,, \tag{2.15}$$

from which a solution for θ_1 can be obtained as

$$\theta_1 = -2a\tan\left(\frac{\mathbf{p}_x + \mathbf{a}_x d_6 + a_7\mathbf{n}_x - (\mathbf{a}_x{}^2 d_6{}^2 + 2d_6(a_7\mathbf{n}_x + \mathbf{p}_x)\mathbf{a}_x + \mathbf{a}_y{}^2 d_6{}^2 + 2d_6(a_7\mathbf{n}_y + \mathbf{p}_y)\mathbf{a}_y + a_7{}^2\mathbf{n}_x{}^2 + a_7{}^2\mathbf{n}_y{}^2 + 2a_7\mathbf{n}_x\mathbf{p}_x + 2a_7\mathbf{n}_y\mathbf{p}_y - d_2{}^2 + \mathbf{p}_x{}^2 + \mathbf{p}_y{}^2}{a_7\mathbf{n}_y + \mathbf{a}_y d_6 - d_2 + \mathbf{p}_y}\right) \tag{2.16}$$

Now that θ_1 is solved, θ_5 can be obtained directly from equation (2.13) element [3,3] as

$$\theta_5 = a\sin(\mathbf{a}_x \sin(\theta_1) - \mathbf{a}_y \cos(\theta_1) \tag{2.17}$$

and θ_6 from elements [3,1] and [3,2] as

$$\theta_6 = a\tan2(-\mathbf{n}_x \sin(\theta_1) + \mathbf{n}_y \cos(\theta_1), -\mathbf{s}_x \sin(\theta_1) + \mathbf{s}_y \cos(\theta_1)) \tag{2.18}$$

In order to obtain angle θ_{234}, equation (2.13) is postmultiplied by $\left[\mathbf{A}_7^6\right]^{-1}\left[\mathbf{A}_6^5\right]^{-1}\left[\mathbf{A}_5^4\right]^{-1}$ leading to

$$\left[\mathbf{A}_1^0\right]^{-1}\mathbf{T}_7^0\left[\mathbf{A}_7^6\right]^{-1}\left[\mathbf{A}_6^5\right]^{-1}\left[\mathbf{A}_5^4\right]^{-1} = \mathbf{A}_2^1\mathbf{A}_3^2\mathbf{A}_4^3 \tag{2.19}$$

Taking elements [1,1] and [2,1] respectively yields

$$\cos_{234} = ((-c_6\mathbf{s}_x - s_6\mathbf{n}_x)c_1 - s_1(c_6\mathbf{s}_y + s_6\mathbf{n}_y))s_5 - c_5(\mathbf{a}_x c_1 + \mathbf{a}_y s_1) \tag{2.20}$$

$$\sin_{234} = -s_5 c_6 \mathbf{s}_z - s_5 s_6 \mathbf{n}_z - \mathbf{a}_z c_5 \tag{2.21}$$

where $c_1 = \cos(\theta_1)$, $s_1 = \sin(\theta_1)$, $c_{234} = \cos(\theta_2 + \theta_3 + \theta_4)$ and so forth.

Now angle θ_{234} can be solved with the two-argument arctangent as

$$\theta_{234} = a\tan2(\sin_{234}, \cos_{234}) \tag{2.22}$$

Angle θ_3 can be obtained from equation (2.13) elements [1,4] and [2,4]. Modifying the element [1,4] from the right-hand side of the equation by expanding and trigonometrically combining it and setting it equal with the left-hand side of the equation yields

$$\mathbf{p}_x c_1 - a_1 + \mathbf{p}_y s_1 = a_4 c_{234} + d_6 c_5 c_{234} + a_7 c_6 s_{234} + a_7 s_5 s_6 c_{234} + a_3 c_{23} + a_2 c_2 - d_5 s_{234} \tag{2.23}$$

Rearranging equation (2.23) we get equations for the left- and right-hand sides as

$$\mathbf{T}_7^1 left[1,4] = \mathbf{p}_x c_1 - a_1 + \mathbf{p}_y s_1 - a_4 c_{234} - d_6 c_5 c_{234} - a_7 c_6 s_{234} - a_7 s_5 s_6 c_{234} + d_5 s_{234} \tag{2.24}$$

$$\mathbf{T}_7^1 right[1,4] = a_3 c_{23} + a_2 c_2 \tag{2.25}$$

In a similar manner as above, element [2,4] of equation (2.13) yields

$$\mathbf{p}_z - d_1 = a_3 s_{23} + a_7 s_5 s_6 s_{234} + d_6 c_5 s_{234} - a_7 c_6 c_{234} + a_2 s_2 + d_5 c_{234} + a_4 s_{234} \tag{2.26}$$

Rearranging equation (2.26) we get equations for the left- and right-hand sides as

$$\mathbf{T}_7^1 left[2,4] = \mathbf{p}_z - d_1 - a_7 s_5 s_6 s_{234} - d_6 c_5 s_{234} + a_7 c_6 c_{234} - d_5 c_{234} - a_4 s_{234} \tag{2.27}$$

$$\mathbf{T}_7^1 right[2,4] = a_3 s_{23} + a_2 s_2 \tag{2.28}$$

Now, $\cos(\theta_3)$ can be obtained by squaring and summing both sides of the equation and then subtracting them. This leads to

$$\cos(\theta_3) = ((\mathbf{T}_7^1 right[1,4]^2 + \mathbf{T}_7^1 right[2,4]^2) - (\mathbf{T}_7^1 left[1,4]^2 + \mathbf{T}_7^1 left[2,4]^2)) \tag{2.29}$$

$\mathrm{Sin}(\theta_3)$ can be obtained with the trigonometric identity

$$\sin(\theta)^2 + \cos(\theta)^2 = 1 \tag{2.30}$$

which leads to

$$\sin(\theta_3) = \sqrt{1 - \cos(\theta_3)^2} \tag{2.31}$$

Now angle θ_3 can be expressed with the two-argument arctangent as

$$\theta_3 = -a\tan2(\sin(\theta_3), \cos(\theta_3)) \tag{2.32}$$

For solving angle θ_2, elements [1,4] and [2,4] from equation (2.13) are constructed utilizing equations (2.24), (2.25), (2.27) and (2.28). This leads to two equations

$$\mathbf{T}_7^1[1,4] = \mathbf{T}_7^1 left[1,4] - \mathbf{T}_7^1 right[1,4] \tag{2.33}$$

$$\mathbf{T}_7^1[2,4] = \mathbf{T}_7^1 left[2,4] - \mathbf{T}_7^1 right[2,4] \tag{2.34}$$

Now, $\sin(\theta_2)$ and $\cos(\theta_2)$ can be simultaneously solved as

$$\sin(\theta_2) = \frac{-\cos(\theta_3)a_3\mathbf{T}_7^1 left[2,4] - a_2\mathbf{T}_7^1 left[2,4] + \sin(\theta_3)a_3\mathbf{T}_7^1 left[1,4]}{{a_3}^2 + {a_2}^2 + 2a_3\cos(\theta_3)a_2} \tag{2.35}$$

$$\cos(\theta_2) = \frac{\cos(\theta_3)a_3\mathbf{T}_7^1 left[1,4] + a_2\mathbf{T}_7^1 left[1,4] + \sin(\theta_3)a_3\mathbf{T}_7^1 left[2,4]}{{a_3}^2 + {a_2}^2 + 2a_3\cos(\theta_3)a_2} \tag{2.36}$$

Now angle θ_2 can be expressed using equations (2.35) and (2.36) as

$$\theta_2 = -a\tan\left(\frac{\sin(\theta_2)}{\cos(\theta_2)}\right) \tag{2.37}$$

Finally, angle θ_4 is obtained as

$$\theta_4 = \theta_{234} - \theta_3 - \theta_2 \tag{2.38}$$

In conclusion, a closed-form solution for the inverse kinematics problem could be solved. However, when utilizing the inverse kinematics solution with the excavator model, some restrictions must be applied on the joint angles, as the kinematic construction limits the admissible joint ranges.

2.2.2 Euler XYZ angles

The rotation matrix **R** (2.5) defined in section 2.1 is characterized by nine elements, which are not independent but related by six constraints due to the orthogonality conditions mentioned in equation (2.6). This means that in order to sufficiently describe the orientation of a rigid body in space, only three parameters are needed. These three parameters make up a three-angle representation of the orientation, namely Euler XYZ-angles, which are widely used in robotics and especially in the aeronautical field [2]. The XYZ-angles are also known as the Roll-Pitch-Yaw angles.

The Roll-Pitch-Yaw angles can be defined as

$$\phi = [\varphi \quad \upsilon \quad \psi]^T \tag{2.39}$$

where the angle φ describes a rotation about axis x (roll), angle υ a rotation about axis y (pitch) and ψ a rotation about axis z (yaw).

The orientation of the frame with respect to a fixed reference frame is obtained by multiplying the matrices of elementary rotations, which yields

$$\mathbf{R}(\phi) = \mathbf{R}_z(\psi)\mathbf{R}_y(\upsilon)\mathbf{R}_x(\varphi) = \begin{bmatrix} c_\psi c_\upsilon & c_\psi s_\upsilon s_\varphi - s_\psi c_\varphi & c_\psi s_\upsilon c_\varphi + s_\varphi s_\psi \\ s_\psi c_\upsilon & s_\psi s_\upsilon s_\varphi + c_\psi c_\varphi & s_\psi s_\upsilon c_\varphi - c_\psi s_\varphi \\ -s_\upsilon & c_\upsilon s_\varphi & c_\upsilon c_\varphi \end{bmatrix} \tag{2.40}$$

The inverse solution to a given rotation matrix can be obtained by comparing it with equation (2.40) as

$$\begin{aligned} \varphi &= a\tan2(\mathbf{s}_z, \mathbf{a}_z) \\ \upsilon &= -a\sin(\mathbf{n}_z) \\ \psi &= a\tan2(\mathbf{n}_y, \mathbf{n}_x), \end{aligned} \tag{2.41}$$

provided that cos(υ) ≠ 0 [5]. This Roll-Pitch-Yaw representation is valid in the majority of cases, but some rotations may lead to two axes being coincident resulting in one dimension diminishing entirely. This situation is called gimbal lock and it can be overcome with a four-parameter representation called the unit quaternion [6].

2.2.3 Unit quaternion

Unit quaternions are a compact method of depicting an angle and rotation about an axis. They are comprised of two components: a scalar part η and a vector part ε, which consists of three imaginary axes ε_x, ε_y and ε_z making it an extension to complex numbers. Euler parameters can be defined as unit quaternions as

$$Q = \{\eta, \varepsilon\}, \tag{2.42}$$

where $\eta = \cos\frac{\upsilon}{2}$ and $\varepsilon = \frac{\upsilon}{2}\mathbf{r}$.

The four parameters of the unit quaternion are constrained by the condition

$$\eta^2 + \varepsilon_x{}^2 + \varepsilon_y{}^2 + \varepsilon_z{}^2 = 1 \tag{2.43}$$

The forward kinematics equation gives the rotation of the end effector with respect to the reference base frame. However, it is useful to transform the rotation matrix **R** to quaternions due to their superior properties. In the transformation, the following result is useful.

$$\eta = \frac{1}{2}\sqrt{\mathbf{n}_x + \mathbf{s}_y + \mathbf{a}_z + 1}$$

$$\varepsilon = \frac{1}{2}\begin{bmatrix} signum(\mathbf{s}_z - \mathbf{a}_y)\sqrt{\mathbf{n}_x - \mathbf{s}_y - \mathbf{a}_z + 1} \\ signum(\mathbf{a}_x - \mathbf{n}_z)\sqrt{\mathbf{s}_y - \mathbf{a}_z - \mathbf{n}_x + 1} \\ signum(\mathbf{n}_y - \mathbf{s}_x)\sqrt{\mathbf{a}_z - \mathbf{n}_x - \mathbf{s}_y + 1} \end{bmatrix}, \tag{2.44}$$

where signum(x) = 1 for $x \geq 0$ and signum(x) = -1 for $x \leq 0$. Assuming that $\eta > 0$ corresponds to an angle $\upsilon \in [-\pi,\pi]$, meaning that any rotation can be described. [2]

In many cases, it is more intuitive to express the rotation as Euler XYZ angles. The transformation from unit quaternions to Roll-Pitch-Yaw angles can be done as

$$\begin{bmatrix} \varphi \\ \upsilon \\ \psi \end{bmatrix} = \begin{bmatrix} a\tan 2(\mathbf{q}_y\mathbf{q}_z + \mathbf{q}_w\mathbf{q}_x, 0.5 - (\mathbf{q}_x{}^2 + \mathbf{q}_y{}^2)) \\ a\sin(-2*(\mathbf{q}_x\mathbf{q}_z - \mathbf{q}_w\mathbf{q}_y)) \\ a\tan 2(\mathbf{q}_x\mathbf{q}_y + \mathbf{q}_w\mathbf{q}_z, 0.5 - (\mathbf{q}_y{}^2 + \mathbf{q}_z{}^2)) \end{bmatrix} \tag{2.45}$$

where $\mathbf{q}_w$ is the scalar part and $\mathbf{q}_x$, $\mathbf{q}_y$, $\mathbf{q}_z$ the vector part of the unit quaternion. [14]

3. DYNAMICS

The dynamic model of a manipulator has an essential role in computer simulations of motion, analysis of manipulator structures and in the design of controller algorithms. By running computer simulations, different controller schemes and trajectory planning sequences can be tested without having a physical machine at disposal. The manipulator kinematics describe the mechanical construction and movement in free space where no outside factors affect it. When elements such as friction (internal force) and gravitational and contact forces (external forces) are taken into account, the relationship between these factors and the manipulator are realized by means of model dynamics, which is the derivation of equation of motion. In other words, dynamics describe why and how a motion related to the kinematic structure occurs when forces and moments are applied to the system. [2]

Dynamics can be divided in two major problems. In the first problem, a specific motion of the manipulator is desired, more accurately the movement of each individual link, and the forces and moments required to achieve this motion are under consideration. This approach is called direct dynamics and it is easier to solve when the equations of motion are known since it requires differentiating of kinematic equations. Direct dynamics include the study of static conditions, which reduces the problem to solving forces that maintain a stationary position of the manipulator. Direct dynamics do not however limit to static conditions only. A very important aspect of the direct dynamics is calculating the forces needed to move the end-effector on a given path from an initial configuration to the final configuration with a predetermined time history. [5]

In the second dynamics problem, the applied forces and moments are known and the consequent motion of the manipulator is under consideration. This approach is called inverse dynamics and it is more difficult to solve compared to direct dynamics due to the need of integration of equations of motion. The inverse dynamics problem has several interesting applications, as it essentially is a prediction of motion given a certain initial state to each link of the manipulator. [5]

The derivation of the dynamic model of a manipulator utilizes a number of different methods and mathematical tools. A select portion of these will be discussed in detail in sections 3.1 and 3.2.

3.1 Jacobian matrix

Forward and inverse kinematics establish the relationship between the end-effector position and orientation and the manipulator joint variables. However, they only provide static state conditions of the manipulator; hence, a different approach is required in order to

examine dynamic cases. This can be achieved with the Jacobian matrix, which is a very important and extremely versatile tool in both manipulator kinematics and dynamics. The Jacobian matrix can be constructed either in a geometric form or by means of differentiation of the forward kinematics equation with respect to the joint variables. The geometric Jacobian depends on the manipulator configuration while in order to express the analytical Jacobian via differentiation, the end-effector must be expressed with reference to a minimal representation, namely Euler angles, in the operational space.

With the Jacobian matrix, it is possible to link the manipulator joint velocities with the corresponding end-effector linear and angular velocities. In addition, the Jacobian matrix can be used to map the forces applied to the end-effector and the resulting torques at the manipulator joints. Also, redundancy analyzation and singular configuration identification as well as inverse kinematics algorithm determination are possible with the Jacobian matrix.

The forward kinematics equations formulated in section 2.1.2 describe the transformation between the joint space and the Cartesian space. By differentiating the position vector, the relationship between the end-effector linear velocities and the joint angular velocities can be determined as time derivatives of the forward kinematics equations as:

$$\dot{\boldsymbol{x}} = \boldsymbol{J}(\theta)\dot{\theta}, \tag{3.1}$$

where the Jacobian $\mathbf{J}(\theta)$ is defined as:

$$\boldsymbol{J}(\theta) = \begin{bmatrix} \frac{\partial x_1}{\partial \theta_1} & \cdots & \frac{\partial x_1}{\partial \theta_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial x_m}{\partial \theta_1} & \cdots & \frac{\partial x_m}{\partial \theta_n} \end{bmatrix} \tag{3.2}$$

3.2 Lagrangian dynamics equations

The dynamics of a manipulator depend on its inertial and kinematic properties. These properties can be described as a set of nonlinear, second-order differential equations, which can be derived using Lagrangian dynamic formulation. The Lagrange's equations are based on the kinetic and potential energy properties in the mechanical system with respect to the individual joint variables $\mathbf{q}_i$. These equations can be calculated in closed form, which allows the properties of the system to be analyzed in detail. The joint variables $\mathbf{q}_i$ are expressed in generalized coordinates, which allows the equations of motion to be expressed in terms of generalized coordinates and the external forces applied to the system as components along these generalized coordinates.

The Lagrangian L is the difference between the kinetic and potential energy in the system and it can be defined as

$$\boldsymbol{L}(\boldsymbol{q},\dot{\boldsymbol{q}}) = \boldsymbol{K}(\boldsymbol{q},\dot{\boldsymbol{q}}) - \boldsymbol{P}(\boldsymbol{q}), \tag{3.3}$$

where $\boldsymbol{K}$ is the kinetic energy and $\boldsymbol{P}$ is the potential energy of the system in generalized coordinates.

The Lagrange's equations for the equations of motion can be defined as

$$\frac{d}{dt}\frac{\partial \boldsymbol{L}}{\partial \dot{\boldsymbol{q}}_i} - \frac{\partial \boldsymbol{L}}{\partial \boldsymbol{q}_i} = \tau_i, \tag{3.4}$$

where τ_i, $i=1...n$ is the external force/torque corresponding to the i^{th} joint of the manipulator.

3.3 Dynamics of the excavator

The effect of each link of the manipulator to the equations of motion depend on the position vector of each link's center of gravity with respect to the generalized coordinate frame. The center of gravity for the boom, stick and the combination of the tiltrotator and the bucket are shown in figure 5 along with the position vectors connecting them to the related manipulator link joint.

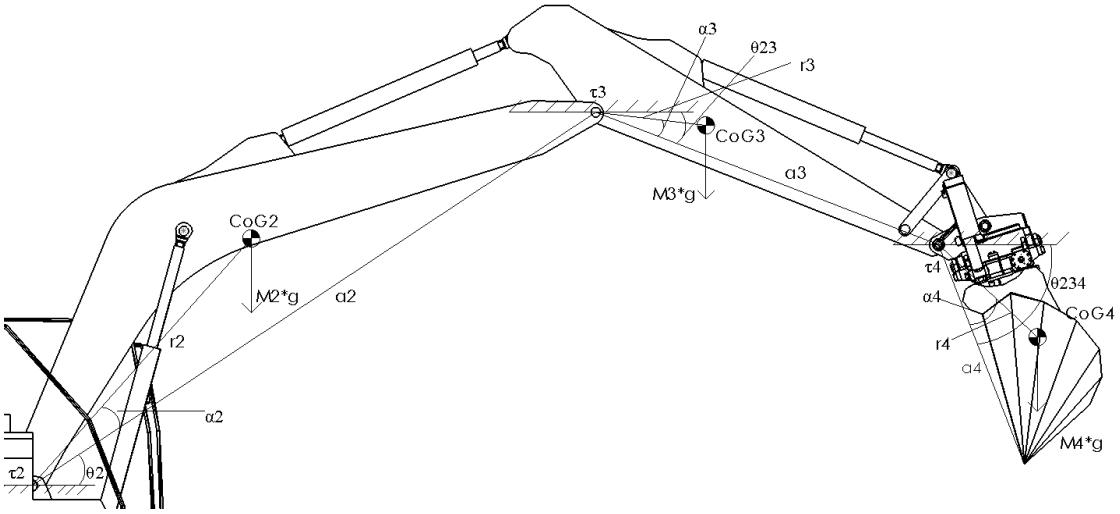

Figure 5. Excavator arm schematics for links' centers of gravity

From figure 5, the kinematic equations related to each link of the manipulator can be derived for the equations of motion. The potential energy of the manipulator can be written as

$$\boldsymbol{P} = \sum_{i=1}^{n} m_i g\, r_{ci}, \tag{3.5}$$

where m_i is the mass of link i, g is the gravitational acceleration and r_{ci} is the position vector from the origin to the center of mass of link i.

For the sake of simplicity, the fifth and sixth degree of freedom (bucket tilt and rotation) have been omitted from the calculations and the tiltrotator and bucket are considered as one rigid link. The resulting torque $\boldsymbol{\tau}_{4g}$ at joint four from the gravitational components can be expressed as

$$\tau_{4g} = m_4 g r_4 \cos(\theta_{234} - \alpha_4), \tag{3.6}$$

where $\theta_{234} = \theta_2 + \theta_3 + \theta_4$.

Joint torque $\boldsymbol{\tau}_{3g}$ can be expressed in a similar manner as

$$\tau_{3g} = \tau_{4g} + m_4 g a_3 \cos(\theta_{23}) + m_3 g r_3 \cos(\theta_{23} - \alpha_3) \tag{3.7}$$

and torque $\boldsymbol{\tau}_{2g}$ as

$$\tau_{2g} = \tau_{3g} + (m_4 + m_3) g a_2 \cos(\theta_2) + m_2 g r_2 \cos(\theta_2 + \alpha_2). \tag{3.8}$$

In the examination of the manipulator dynamics, the first joint (cabin slew) is considered to be fixed putting the resulting degrees of freedom on the *xz*-plane. This allows the reference frame origin to be set at the origin of the boom link. For the calculation of the kinetic energy $\boldsymbol{K}$, the center of gravity of each link has to be expressed in the generalized coordinates with respect to the origin, which was previously defined. The center of gravity for the boom, stick and bucket are

$$\begin{bmatrix} \mathbf{x}_{bo} \\ \mathbf{z}_{bo} \end{bmatrix} = \begin{bmatrix} r_2 \cos(\theta_2 + \alpha_2) \\ r_2 \sin(\theta_2 + \alpha_2) \end{bmatrix} \tag{3.9}$$

$$\begin{bmatrix} \mathbf{x}_{st} \\ \mathbf{z}_{st} \end{bmatrix} = \begin{bmatrix} a_2 \cos(\theta_2) + r_3 \cos(\theta_{23} - \alpha_3) \\ a_2 \sin(\theta_2) + r_3 \sin(\theta_{23} - \alpha_3) \end{bmatrix} \tag{3.10}$$

$$\begin{bmatrix} \mathbf{x}_{bu} \\ \mathbf{z}_{bu} \end{bmatrix} = \begin{bmatrix} a_2 \cos(\theta_2) + a_3 \cos(\theta_{23}) + r_4 \cos(\theta_{234} - \alpha_4) \\ a_2 \sin(\theta_2) + a_3 \sin(\theta_{23}) + r_4 \sin(\theta_{234} - \alpha_4) \end{bmatrix} \tag{3.11}$$

and therefore, the velocity of the center of gravity for each link is

$$\begin{bmatrix} \dot{\mathbf{x}}_{bo} \\ \dot{\mathbf{z}}_{bo} \end{bmatrix} = \begin{bmatrix} -r_2 \sin(\theta_2 + \alpha_2)\dot{\theta}_2 \\ r_2 \cos(\theta_2 + \alpha_2)\dot{\theta}_2 \end{bmatrix} \tag{3.12}$$

$$\begin{bmatrix} \dot{\mathbf{x}}_{st} \\ \dot{\mathbf{z}}_{st} \end{bmatrix} = \begin{bmatrix} (-a_2 \sin(\theta_2) - r_3 \sin(\theta_{23} - \alpha_3))\dot{\theta}_2 - r_3 \sin(\theta_{23} - \alpha_3)\dot{\theta}_3 \\ (a_2 \cos(\theta_2) + r_3 \cos(\theta_{23} - \alpha_3))\dot{\theta}_2 + r_3 \cos(\theta_{23} - \alpha_3)\dot{\theta}_3 \end{bmatrix} \tag{3.13}$$

$$\begin{bmatrix} \dot{\mathbf{x}}_{bu} \\ \dot{\mathbf{z}}_{bu} \end{bmatrix} = \begin{bmatrix} (-a_2 \sin(\theta_2) - a_3 \sin(\theta_{23}) - r_4 \sin(\theta_{234} - \alpha_4))\dot{\theta}_2 - (a_3 \sin(\theta_{23}) + r_4 \sin(\theta_{234} - \alpha_4))\dot{\theta}_3 - r_4 \sin(\theta_{234} - \alpha_4)\dot{\theta}_4 \\ (a_2 \cos(\theta_2) + a_3 \cos(\theta_{23}) + r_4 \cos(\theta_{234} - \alpha_4))\dot{\theta}_2 + (a_3 \cos(\theta_{23}) + r_4 \cos(\theta_{234} - \alpha_4))\dot{\theta}_3 + r_4 \cos(\theta_{234} - \alpha_4)\dot{\theta}_4 \end{bmatrix}. \tag{3.14}$$

Now the kinetic energy of the manipulator can be expressed as the sum of the kinetic energy of the individual links as $\boldsymbol{K} = \boldsymbol{K}_{bo} + \boldsymbol{K}_{st} + \boldsymbol{K}_{bu}$:

$$\boldsymbol{K} = \frac{\mathbf{1}}{\mathbf{2}} m_2 \left(\dot{x}_{bo}{}^2 + \dot{z}_{bo}{}^2 \right) + \frac{\mathbf{1}}{\mathbf{2}} m_3 \left(\dot{x}_{st}{}^2 + \dot{z}_{st}{}^2 \right) + \frac{\mathbf{1}}{\mathbf{2}} m_4 \left(\dot{x}_{bu}{}^2 + \dot{z}_{bu}{}^2 \right) \tag{3.15}$$

and the manipulator kinetic energy can be written with the Lagrangean L as partial derivatives with respect to each joint variable as:

$$\boldsymbol{K} = \frac{d}{dt} \left(\frac{\partial \boldsymbol{L}}{\partial \dot{\theta}_i} \right). \tag{3.16}$$

The resulting kinetic energy consists of the inertial components of the manipulator dynamics as well as the effects of Coriolis and centrifugal terms. However, the internal forces such as friction in the manipulator joints, valves and actuators is not taken into account.

Finally, the manipulator dynamics derived with the Lagrangean equations of motion can be expressed in the joint space as a decoupled, second-order differential equation as

$$\tau = D(\boldsymbol{q})\ddot{\boldsymbol{q}} + C(\boldsymbol{q}, \dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + G(\boldsymbol{q}), \tag{3.17}$$

where D(**q**) is the inertia matrix, C($\boldsymbol{q}$, $\dot{\boldsymbol{q}}$)$\dot{\boldsymbol{q}}$ the vector of Coriolis and centrifugal terms, G(**g**) the gravity torque vector, **q** the joint-variable vector and **τ** the joint torque vector.

3.4 Joint torque and actuator force mapping

With manipulator dynamics, the applied forces and torques to the manipulator can be in a number of different coordinate frames. In order to properly direct the forces and torques to the appropriate target, transformations between coordinate frames are necessary. With the Jacobian transpose matrix, which was discussed earlier, forces in the operational space can be converted to the joint space and with the pseudoinverse of the transpose back to the operational space. However, it is equally necessary to convert between the joint space and the actuator space from time to time. In order to do so, a relation must be constructed between the actuator length and the corresponding joint angle.

3.4.1 Joint variable and actuator length relation

The relation between the cylinder length $\mathbf{z}_2$ and the joint angle $\mathbf{q}_2$ is demonstrated in figure 6.

Angles β_1 and β_2 can be calculated using the dimensions found in the CAD-model of the excavator. Using the cosine law, we get for the actuator length $\mathbf{z}_2$:

$$\boldsymbol{z}_2 = \sqrt{L11^2 + L12^2 - 2L11L12\cos(\boldsymbol{q}_2)}, \tag{3.18}$$

where $\mathbf{q}_2$ is $\beta_1 + \beta_2 + \theta_2$.

Joint variable $\mathbf{q}_2$ can be expressed as a function of $\mathbf{z}_2$ as:

$$\boldsymbol{q}_2 = \text{acos}\left(\frac{L11^2+L12^2-\boldsymbol{z}_2{}^2}{2L11L12}\right), \tag{3.19}$$

from which the relationship between the joint angle and the actuator stroke can be derived.

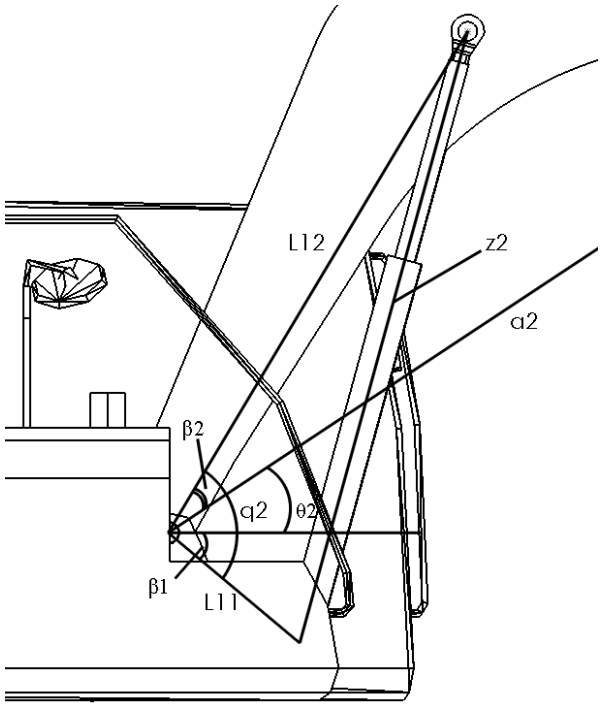

Figure 6. Boom cylinder length and joint angle mapping

In a similar manner as above, the relation between the actuator length $\mathbf{z}_3$ and the corresponding joint angle $\mathbf{q}_3$ is shown in figure 7.

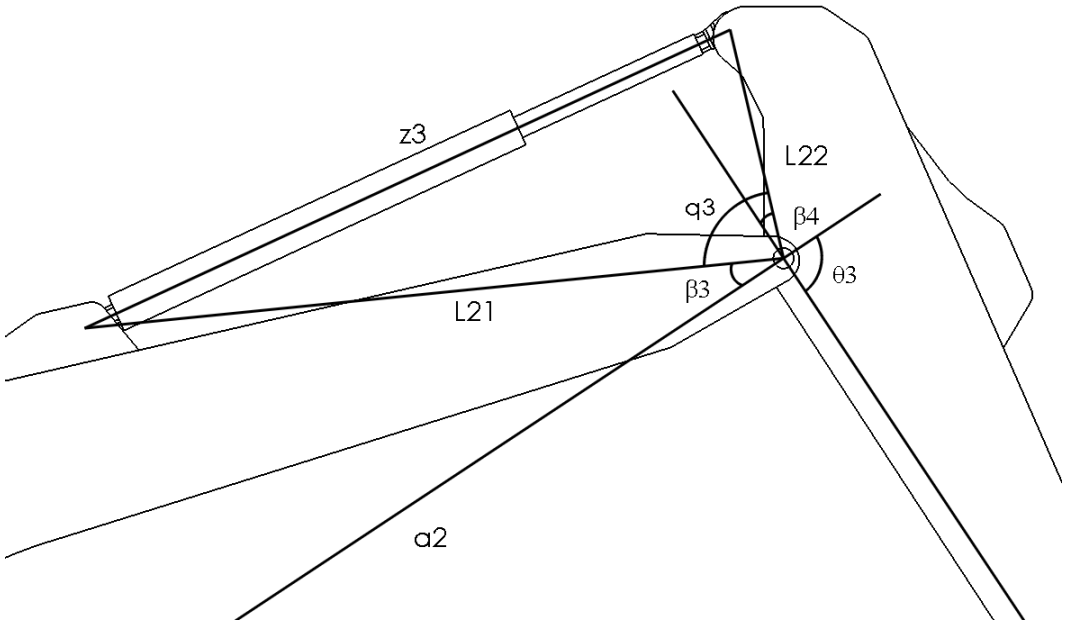

Figure 7. Stick cylinder length and joint angle mapping

As with above, angles β_3 and β_4 can be calculated using the dimensions found in the CAD-model of the excavator. Using the cosine law, we get for the actuator length $\mathbf{z}_3$:

$$\boldsymbol{z}_3 = \sqrt{L21^2 + L22^2 - 2L21L22\cos(\boldsymbol{q}_3)}, \tag{3.20}$$

where $\mathbf{q}_3$ is β_4 - β_3 - θ_3.

Joint variable $\mathbf{q}_3$ can be expressed as a function of $\mathbf{z}_3$ as:

$$\boldsymbol{q}_3 = \operatorname{acos}\left(\frac{L21^2+L22^2-\boldsymbol{z}_3{}^2}{2L21L22}\right). \tag{3.21}$$

The relation between actuator length $\mathbf{z}_4$ and joint variable $\mathbf{q}_4$ is considerably more difficult to derive since they are connected via a four-bar linkage. The four-bar linkage consisting of the tiltrotator, manipulator stick and the hydraulic cylinder is shown in figure 8.

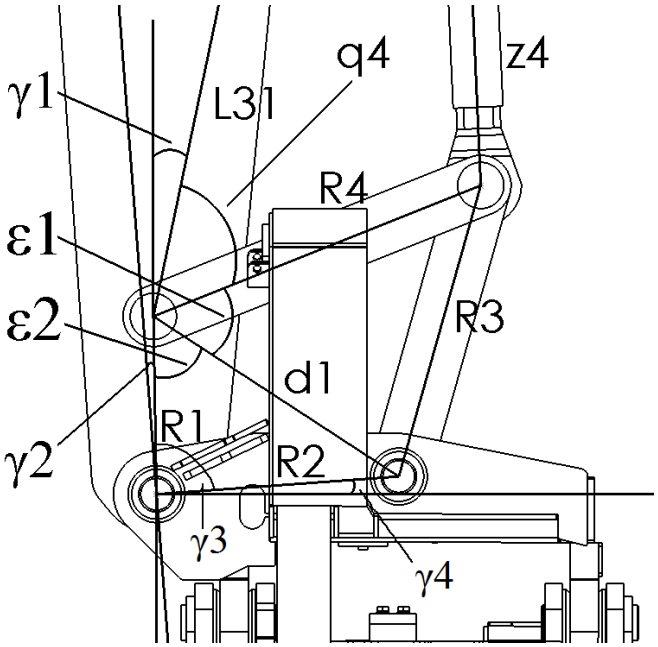

Figure 8. Bucket cylinder length and joint angle mapping

The actuator length $\mathbf{z}_4$ as a function of the joint variable $\mathbf{q}_4$ is obtained with the cosine law as:

$$\boldsymbol{z}_4 = \sqrt{L31^2 + R4^2 - 2L31R4\cos(\boldsymbol{q}_4)}, \tag{3.22}$$

and reversed, the joint variable $\mathbf{q}_4$ as a function of the actuator length $\mathbf{z}_4$ can be expressed as:

$$\boldsymbol{q}_4 = \operatorname{acos}\left(\frac{L31^2+R4^2-\boldsymbol{z}_4{}^2}{2L31R4}\right). \tag{3.23}$$

Now, the joint variable $\mathbf{q}_4$ must be expressed as a function of the joint angle θ_4. From figure 8, angles γ_2 and γ_4, which can be calculated using the dimensions from the CAD-model, are constant resulting in the four-bar linkage angle γ_3 to change in unison with angle θ_4 about the joint connecting the stick and the tiltrotator. Having solved angle γ_3; the diagonal d_1 of the four-bar linkage can be obtained with the cosine law as:

$$d_1 = \sqrt{R1^2 + R2^2 - 2R1R2\cos(\gamma_3)}. \tag{3.24}$$

With the diagonal d_1, angles ε_1 and ε_2 can be obtained as:

$$\varepsilon_1 = acos\left(\frac{d_1{}^2+R4^2-R3^2}{2d1R4}\right) \tag{3.25}$$

$$\varepsilon_2 = acos\left(\frac{d_1{}^2+R1^2-R2^2}{2d1R1}\right). \tag{3.26}$$

Lastly, angle γ_1 can be calculated using the dimensions found in the CAD-model. Now the joint variable $\mathbf{q}_4$ can be expressed as a function of θ_4 as:

$$\boldsymbol{q}_4 = \pi + \gamma_2 - \gamma_1 - \varepsilon_1 - \varepsilon_2, \tag{3.27}$$

where angles ε_1 and ε_2 are functions of angle θ_4.

3.4.2 Virtual work principle

Virtual work can be defined as the work done by a real force acting through a virtual displacement or a virtual force acting through a real displacement. Moreover, a virtual displacement is any displacement consistent with the constraints of the structure that satisfy the boundary conditions at the supports, and a virtual force is any system of forces in equilibrium. [8] According to the virtual work principle, dimensions of work of a force and moment of a force are the same even though they are different physical quantities. Neglecting friction in the manipulator joints and in the hydraulic cylinders, the virtual work principle states [7]:

$$\tau d\theta = Fdl, \tag{3.28}$$

where F is the hydraulic force applied by the actuator, τ is the torque applied by the actuator about the corresponding joint and θ and l are the joint angle and cylinder displacement respectively. From equation (3.28) the equation for the force can be derived as:

$$F = \tau J(\theta), \tag{3.29}$$

where the scalar cylinder Jacobian is defined as $J(\theta) = d\theta/dl$.

Equations (3.19), (3.21) and (3.23) derived in section 3.3.1 express the relation between the joint angle and the cylinder stroke for each joint-cylinder pair. By differentiating the equations, the cylinder Jacobian $d\theta/dl$ is obtained as:

$$\mathbf{J}(\theta) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & a2 & 0 & 0 \\ 0 & 0 & a3 & 0 \\ 0 & 0 & 0 & a4 \end{bmatrix}, \tag{3.30}$$

where

$$\begin{aligned} a_2 &= \frac{z_2}{L11L12\sqrt{1-\frac{1}{4}\frac{(L11^2+L12^2-z_2{}^2)^2}{L11^2L12^2}}} \\ a_3 &= \frac{z_3}{L21L22\sqrt{1-\frac{1}{4}\frac{(L21^2+L22^2-z_3{}^2)^2}{L21^2L22^2}}} \\ a_4 &= \frac{z_4}{L31R4\sqrt{1-\frac{1}{4}\frac{(L31^2+R4^2-z_4{}^2)^2}{L31^2R4^2}}} \end{aligned}. \tag{3.31}$$

The element (1,1) in equation (3.30) is 1, because the cabin slew joint was set fixed and it is not taken into account. Moreover, the joint is driven by a hydraulic motor. The cylinder length variables z_2, z_3, z_4 in equation (3.31) are as defined is equations (3.18), (3.20), (3.22) and (3.27).

3.5 External digging force

During a digging operation, the ground exerts a reaction force to the excavator bucket. This reaction force can be expressed as:

$$\boldsymbol{F}_r = k_p \left[k_s bh + \mu N + \varepsilon \left(1 + \frac{V_s}{V_h}\right) bh \sum \Delta x_i \right], \tag{3.32}$$

where k_p and k_s are specific resistance coefficients for the ground material, b and h the width and height of the ground cut slice, μ the friction coefficient between the bucket and ground, N the pressure force of the bucket with the ground, ε the coefficient of resistance in filling the bucket with ground matter during movement, V_s and V_b the volumes of the prism of soil and the bucket repectively, and Δx the increment in meters along the horizontal x-axis. [13]

The resulting reaction force, which is defined to be parallel to the direction of the digging, can be divided into separate horizontal and vertical components as:

$$\boldsymbol{F}_{horiz} = \boldsymbol{F}_r \cos(\theta_{dg} - 0.1) \tag{3.33}$$

$$\boldsymbol{F}_{\boldsymbol{vert}} = \boldsymbol{F}_r \sin(\theta_{dg} - 0.1) \tag{3.34}$$

Consequently, the force components can be expressed as:

$$\boldsymbol{F}_{tangent} = \boldsymbol{F}_r \cos(0.1) \tag{3.35}$$

$$\boldsymbol{F}_{normal} = -\boldsymbol{F}_r \sin(0.1), \tag{3.36}$$

where the force components are conveniently perpendicular. [13]

Finally, the overall resulting dynamics torque can be expressed as:

$$\tau = D(\boldsymbol{q})\ddot{\boldsymbol{q}} + C(\boldsymbol{q}, \dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + G(\boldsymbol{q}) + \boldsymbol{J}(\theta)^T \boldsymbol{F}_r \tag{3.37}$$

4. MANIPULATOR CONTROLLING

In order to control a manipulator and perform desired tasks, a control scheme has to be implemented first. Depending on the level of computational intelligence of the manipulator and which type and how many sensors are installed onboard, the control scheme can be either open or closed-loop type. The simplest case of an open-loop control scheme is a manipulator with no sensors and no controller. Tasks performed on such a manipulator are solely the result of the operator moving each individual actuator via hydraulic valves. The positioning of the end-effector is based on the visual feedback from the manipulator to the operator. This type of manipulator control results in subpar accuracy and low efficiency, but on the other hand, it is relatively robust due to the lack of elements that are at risk to fault.

Adding a controller and sensors to measure the position, velocity and possibly acceleration of the joint variables of the manipulator but still staying true to the open-loop control scheme leads to a system, in which it is possible to follow a predetermined path for the end-effector. The required joint torques that result in the desired path can be calculated with the equations of motion, which were defined in equation (3.31). However, since the control scheme is open-loop and the current state of the manipulator is not used in selecting the control inputs for the joint variables, the performance of the controller is dependent on the accuracy of the measured joint variables and the initial configuration of the manipulator at the start of the desired path.

Since there is no feedback involved, if the desired joint value $\theta_d(t)$ is not equal to the actual joint value $\theta(t)$, the open-loop control scheme will never correct the error. Moreover, in an open-loop control scheme the initial configuration of the manipulator should be exactly the initial point of the desired trajectory for to get even remotely good results. [15]

To overcome the obvious issues with the open-loop control scheme, a feedback loop is introduced to the control scheme resulting in a closed-loop control. A general overview of a closed-loop control scheme is presented in figure 9.

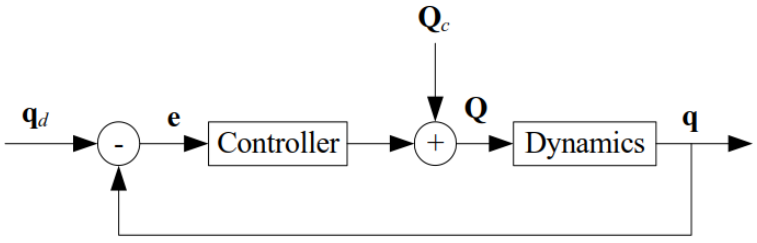

Figure 9. Closed-loop control scheme with feedback control [5]

In a closed-loop control scheme, the required joint torques are calculated in a similar manner as with the open-loop counterpart, but now the error between the desired and actual joint variable values defined as $\mathbf{e} = \boldsymbol{q}_d - \boldsymbol{q}$ is fed to the controller in which a control command signal proportional to the magnitude and time rate of the error is generated. The error vector $\boldsymbol{e}$ is comprised of both the position and velocity of the joint variables, so the command vector $\boldsymbol{Q}$ in figure 9 can be expressed as:

$$\boldsymbol{Q} = \boldsymbol{Q_c} + \boldsymbol{k_d}\dot{\boldsymbol{e}} + \boldsymbol{k_p}\boldsymbol{e}, \tag{4.1}$$

where $\boldsymbol{k}_d$ and $\boldsymbol{k}_p$ are constant control gains and $\boldsymbol{Q}_c$ the feedforward term. [5]

4.1 Path planning

When controlling a manipulator from an initial position to a final position, it is often desired to have the manipulator maneuver between these positions along a certain path in a predetermined time profile. With path or trajectory planning, control inputs for the manipulator are generated so that the desired path is satisfied. Path planning can be done in either the joint space or the operational space. In the joint space, the joint variables are directly specified with respect to time and inverse kinematics equations are not required making it fairly straightforward and simple to calculate joint variable values for the corresponding path. However, visualizing the resulting motion of the manipulator can be difficult in the Cartesian space when defining the path in the joint space. [5] In Cartesian space path planning, a desired trajectory is constructed for the end-effector position and orientation with respect to a time profile. Generating the joint variable values for the path from the Cartesian position and orientation requires the use of the inverse kinematics, which can raise issues if a high requirement for real-time controlling or accuracy is necessary.

For the excavator studied in this thesis, a closed-form solution for the inverse kinematics was derived in section 2.2.1, which allows the joint variable values corresponding to the desired path to be calculated on each simulation step with good accuracy using step sizes as low as few milliseconds. If a closed-form solution for the inverse kinematics is not available for the manipulator under consideration, the joint variable values must be calculated using other methods such as iterative techniques.

Depending on the chosen joint variable tolerances for the accuracy, finding the joint values corresponding to the desired Cartesian path requires several iterations, which leads to an increased sample time for the simulation model.

Generally, it is preferred that the path planning is done in the Cartesian, operational space rather than in the joint space for several advantages. For instance, the description of the desired task for the manipulator is easier to define in the operational space due its intuitive nature compared to the joint space. [2] In addition, restricted areas or obstacles might be

present in the manipulator's workspace. Defining these points are more difficult to compute in the joint space than in the operational space. However, complex rotating sequences may be difficult to generate in the Cartesian space making the path planning in the joint space a reasonable alternative. From here on forth, the majority of the path planning sequences are done in the operational space excluding some exceptions.

The path planning time profile is typically not defined for each point in the geometric path, but as a relation with the total trajectory time, the constraints of the maximum velocities and accelerations, and the assignment of the velocity and acceleration at points of particular interest [2].

4.1.1 Point-to-point motion

When only the initial and final position and orientation of the end-effector are of concern, a so-called point-to-point motion approach is suitable. In this motion, the manipulator moves from the initial to the final pose via an arbitrary path in a given time t_f. No intermediate points can be added between the initial and final pose of the manipulator and no obstacle avoidance can be utilized. The point-to-point motion path planning method is useful for example, when it is desired to move the manipulator to certain predefined configurations such as a storing configuration.

4.1.2 Path motion

A path can be defined as a route consisting of three or more points. When it is desired to perform more elaborate, automated tasks with a manipulator, simple point-to-point motion is not sufficient, but intermediate points must be defined to the path. Adding intermediate points to the path allows achieving desired end-effector position and orientation combinations. Depending on how accurately the end-effector is to be positioned in the Cartesian base coordinate frame, intermediate points can be placed more densely in portions of the path where obstacles are to be avoided or where a high path curvature is expected.

The generated path in the operational space can be described as a series of points as a function of time in the joint space. However, this approach is not very sensible, since it requires each individual point to be calculated using the manipulator inverse kinematics and can unnecessarily burden the controller affecting the simulation step time increasingly. An alternative approach is to use functions such as polynomials to describe the path between two contiguous points of the trajectory. [4] However, selecting a high order polynomial has some drawbacks to it. For starters, initial and final velocities cannot be assigned to the trajectory. Moreover, the numerical accuracy for the computation of the polynomial coefficients deteriorates and the system is more prone to undesired oscillation. In addition, solving of the resulting system of equations gets computationally heavy and the generated path becomes inflexible, meaning that a change in one point in the path

results in the recalculation of all polynomial coefficients. [2] A solution for this problem is to construct the path from a number of low-order interpolating polynomials instead of one high order polynomial and use spline functions around the predefined path points to smoothen out the transition over the path point.

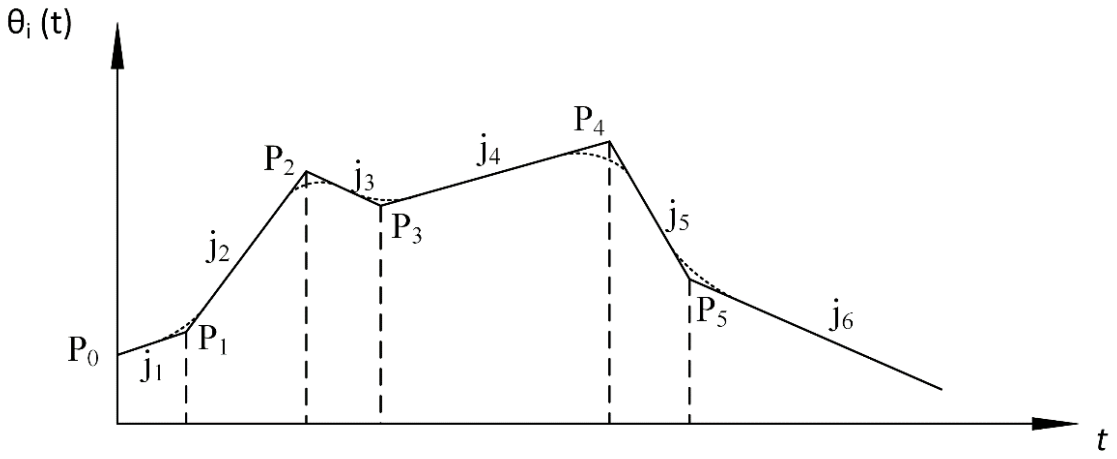

Figure 10. Generated path for revolute joint in joint space

In figure 10 the adjacent interpolating polynomials j_i are connected by the predefined path points P_i by a spline curve function for a smooth transition. The formulation of the path generation is presented in Appendix A.

4.2 Hybrid position/force controller

Manipulator control schemes based solely on the control of the end-effector or joint variable position are appropriate when the trajectory travels through free space and no contact forces are present between the manipulator and the surrounding environment. However, when any external contact is introduced to the manipulator, the position-based controller may become insufficient. [9] In order to compensate for the effects caused by contact forces, a hybrid position/force controller can be implemented as the control scheme.

The hybrid position/force control concept is based on the theory of compliant force and position control. The fundamental idea of the hybrid control is to divide the manipulator's degrees of freedom into natural and artificial constraints. Each manipulation task can be divided into subtasks that are defined by a certain contact situation between the end-effector and the surrounding environment. Natural constraints, which result from the mechanical and geometric characteristics of the task configuration, are then associated to each of these subtasks. For example, an object on a rigid stationary surface cannot go through the surface, which leads to a natural position constraint. If the surface is frictionless, the object in contact with it cannot apply forces tangential to the surface, leading to a natural force constraint. In addition to the natural constraints, additional artificial constraints are introduced in accordance with the natural constraints. The artificial constraints

specify the desired motion or force based on the manipulator trajectory such that artificial position constraints are along tangents, and artificial force constraints are along surfaces, maintaining the consistency with the natural constraints. [10]

Based on the previously defined natural and artificial constraints, three problems can be expressed for the hybrid position/force controller to solve [9]:

- The simultaneous and independent position control of the manipulator along directions in which a natural force constraint exists.
- The simultaneous and independent force control of the manipulator along directions in which natural position constraints exists.
- Devising a scheme to implement the arbitrary mixing of the above-defined control methods along orthogonal degrees of freedom of an arbitrary, Cartesian constraint frame.

The manipulator degrees of freedom can be partitioned into a position-controlled subset and a force-controlled subset using the natural constraints. In order to select which degree of freedom of the manipulator is position controlled and which is force controlled, a diagonal matrix S, called the compliance selectivity matrix can be defined. In the selectivity matrix S, the i^{th} diagonal element is 0 if the degree of freedom's direction is force controlled and 1 if the direction is position controlled. [11]

4.2.1 Explicit force controller

Hybrid position/force controllers can be implemented in several different configurations, which all have unique traits. The method selected in this thesis is an explicit force control based approach developed by M. H. Raibert and J. J. Craig. [10] The general concept of the method is presented in figure 11.

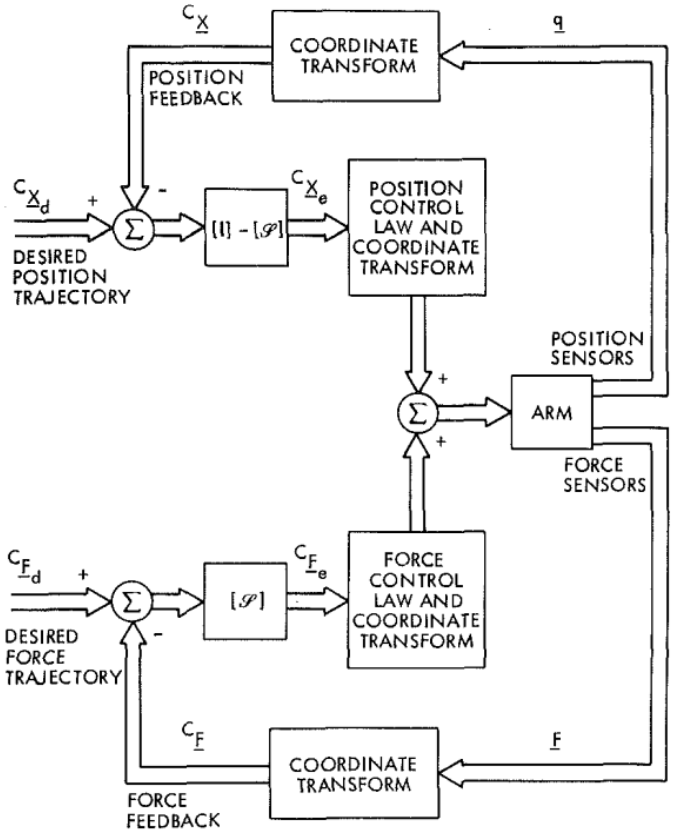

Figure 11. Overview of a hybrid position/force controller [10]

From figure 11, it can be observed that the hybrid controller consists of two complementary, parallel feedback loops. On the top is the upper position feedback loop and on the bottom the lower force feedback loop, which both have their separate and independent sensors and control laws. [10] The feedback signals from the manipulator must be transformed to the earlier mentioned arbitrary Cartesian constraint frame. This constraint frame, formally named $\{C\}$, can be either fixed to the environment or it may move with the end-effector of the manipulator. [9]

For the excavator studied in this thesis, the constraint frame $\{C\}$ was chosen to coincide with the manipulator reference frame. Directions along the x and y axes were chosen to be position controlled and direction along the z axis was chosen as force controlled.

The transformation of the manipulator force feedback signal to the constraint frame is done by taking the cylinder Jacobian derived in section 3.3.2 to transform the cylinder forces to joint torques. The individual cylinder forces can be obtained by measuring the pressure difference between the cylinder chambers. Then the torques can be transformed to Cartesian forces with the inverse of the Jacobian transpose, which allows to use the compliance selectivity matrix S to select the z axis direction for the force portion of the controller. The Simulink implementation of the lower force control loop is presented in figure 12, where the external force is simulated based on equation (3.35) introduced in section 3.5.

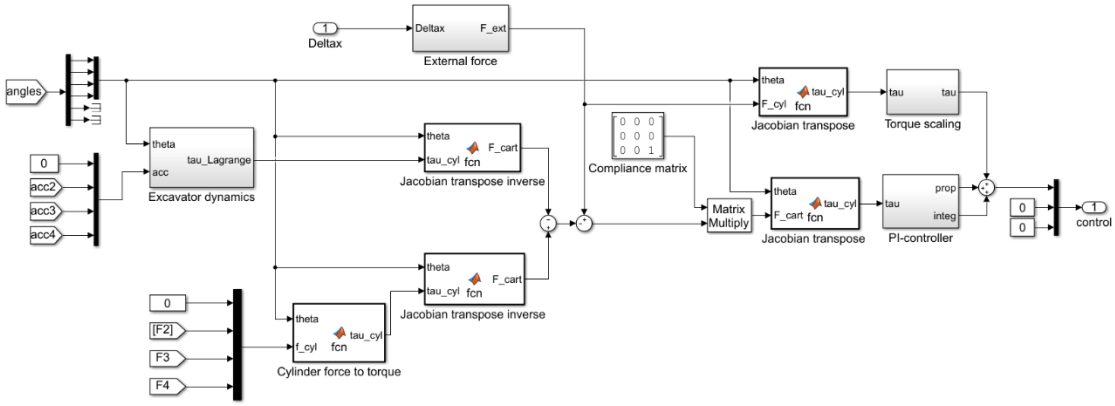

Figure 12. Lower force control loop in hybrid controller

In the upper position feedback control loop, the manipulator position is transformed to the $\{C\}$ frame using the forward kinematics equations derived in section 2.1.2. The velocity feedback can be transformed from the joint space to operational space using the Jacobian matrix. As with the force controller, with the compliance selectivity matrix S the x and y directions are chosen to be position controlled. The upper position control loop is presented in figure 13.

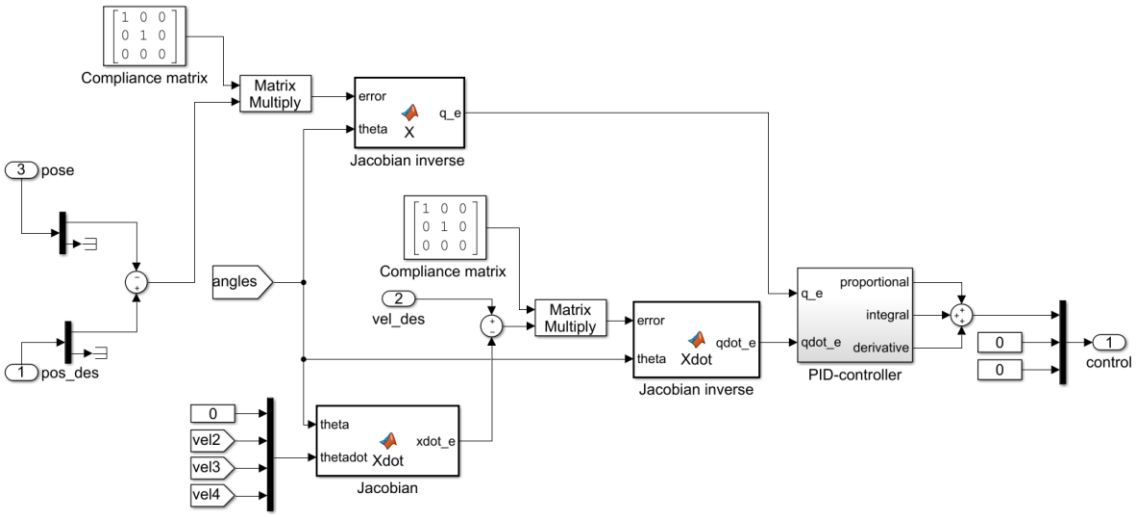

Figure 13. Upper position control loop in hybrid controller

In the $\{C\}$ frame, the position, velocity and force errors can be formulated after which the errors are transformed back to the joint space using the Jacobian inverse for position and velocity and Jacobian transpose for the force. The control laws (4.2) and (4.3) are executed in the joint space.

Based on the controller construction presented in figure 11 and the lower force and upper position controllers in figures 12 and 13, Raibert and Craig proposed in [10] a control law for the upper position feedback loop (4.2) and lower force feedback loop (4.3) as:

$$\boldsymbol{\tau}_p(t) = [K_{pp}]\boldsymbol{q}_e(t) + [K_{pi}]\int[\boldsymbol{q}_e(t)]dt + [K_{pd}]\dot{\boldsymbol{q}}_e(t) \qquad (4.2)$$

$$\boldsymbol{\tau}_f(t) = \tau_{ff}(t) + \left[K_{fp}\right]\tau_e(t) + \left[K_{fi}\right]\int[\tau'_e(t)]dt, \tag{4.3}$$

where $[K_{fp}]$ and $[K_{fi}]$ are proportional and integral force feedback gains, $[K_{pp}]$, $[K_{pi}]$ and $[K_{pd}]$ are proportional, integral and derivative position feedback gains, τ_{ff} is the force feed-forward term, τ'_e is the saturation limited version of τ_e, and τ_p, τ_f are the position and force contribution to the actuator torques respectively. Hence, the total applied torque by the hybrid position/force controller can be expressed as:

$$\boldsymbol{\tau} = \tau_p + \tau_f, \tag{4.4}$$

where the torque $\boldsymbol{\tau}$ is a *(n*x*1)* vector.

The full hybrid position/force controller defined in figures 12 and 13 is presented in figure 14.

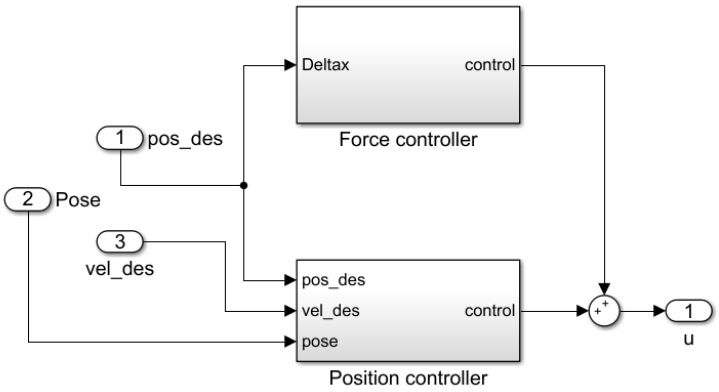

Figure 14. Overview of hybrid position/force controller

4.3 Control methods

Control methods used in the simulation of the excavator studied in this thesis consist of both open and closed-loop control schemes. In the most basic case, the excavator is operated with two two-axis joysticks equipped with function buttons for the bucket tilt and rotation. This setup represents an entry-level excavator with no sensors or controller. The purpose of this open-loop control scheme is to identify and tune the hydraulic valve and actuator parameters in the simulation model to a reasonable extent. However, having insufficient data available of the actual excavator, empiric methods were used in determining some of the parameters. The overview of the open-loop control scheme constructed in MATLAB Simulink is presented in figure 15.

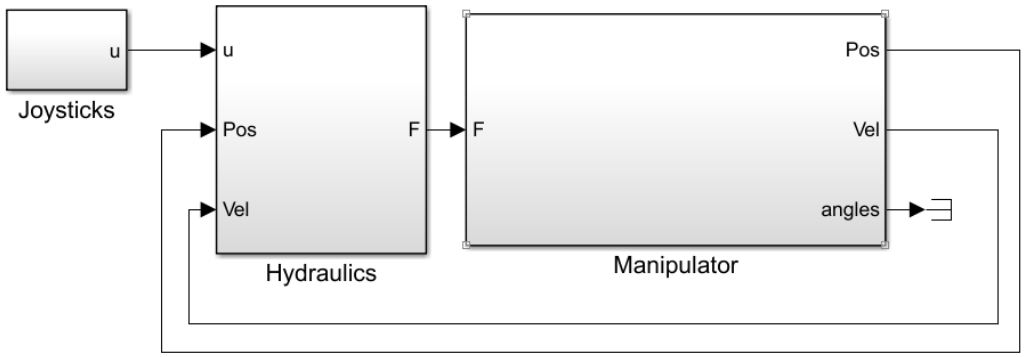

Figure 10. Open-loop control scheme for simulation model

The second control scheme implemented in this thesis is a closed-loop control with a simple PI-controller. Having a controller in the system allows the use of a trajectory generator and enables the use of a Cartesian control scheme, where the end-effector can be maneuvered in the *x, y* and *z* direction of the reference frame with a single joystick direction. The trajectory generator (Appendix A) used in this thesis was developed based on quintic polynomial paths described by Jazar [5]. In figure 16, the trajectory generator produces either a path in the Cartesian space, from which it is transformed into the joint space with the inverse kinematics subsystem, or directly a path in the joint space, in which case the inverse kinematics are not required. In both cases, the PI-controller receives the desired joint angles as an input after which they are compared to the measured joint values from the manipulator. The resulting error is then amplified using a proportional and integrator gain and fed as a control signal to the hydraulic valves. The open-loop control with the joysticks is extended to a closed-loop control due to the added possibility to control the last three joints of the excavator in synchronization. The joystick control is left parallel to the closed-loop controller.

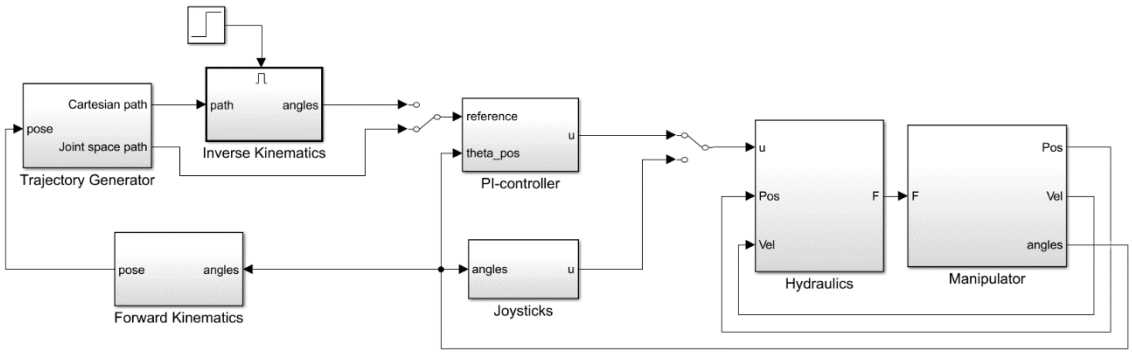

Figure 11. Closed-loop control scheme with PI-controller and inverse kinematics

The third control scheme in this thesis is based on the hybrid position/force controller introduced in section 4.2., in figures 12, 13 and 14. The force feedback loop enables controlling of the manipulator in contact with the environment, which is a common occurrence with excavators. The hybrid controller and the overall simulation model is presented in figure 17.

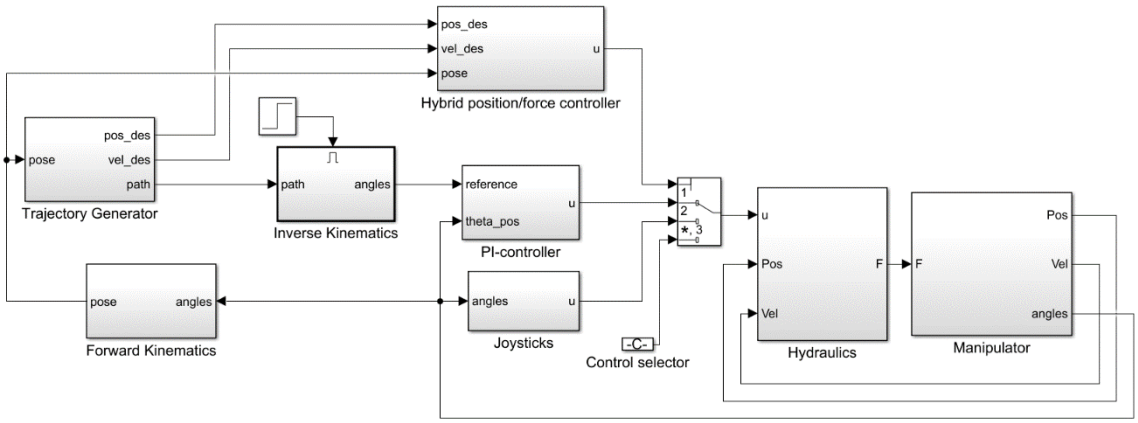

Figure 12. Closed-loop control scheme with hybrid position/force controller

The hybrid position/force controller is positioned parallel to the PI-controller and the closed-loop joystick control scheme. Thus, the desired control scheme can be selected with the multiport switch. The control schemes presented above will be utilized in chapter 5 in the simulation of the manipulator model.

5. SIMULATION RESULTS

The CAD model of the Komatsu PC138US-8 excavator studied in the thesis is presented in Appendix B, and the consequent mechanical model in MATLAB Simulink using the Simmechanics library in Appendix C.

The hydraulics for the simulation model were constructed using the basic Simulink blocks found in the Simulink library browser. For the sake of simplicity, only the key components were constructed and irrelevant components, such as heat exchangers were omitted completely. Moreover, since the focus of this thesis was to study the behavior of the excavator under different control schemes and to model the kinematics and dynamics of the manipulator, the hydraulic pumps have been modeled with a simple constant supply pressure. The developed hydraulic models are presented in Appendix D.

In order to test the created closed-loop control schemes and the accuracy of the excavator simulation model, four key test sequences were created. However, before these can be tested, a sequence for the bucket turn utilizing the tiltrotator was first established.

5.1 Closed-loop tiltrotator sequence

Turning the bucket from the back position to the front position while having a load in the bucket is a complex task, which requires the simultaneous and synchronized use of the last three joints of the excavator. In figure 18, the initial and final pose of the excavator in the bucket turn sequence are presented on the left and right respectively. The tilt degree of freedom is contingent on the angle of the bucket rotation as well as is the bucket pitch. The control scheme for the tiltrotator sequence is presented in Appendix E.

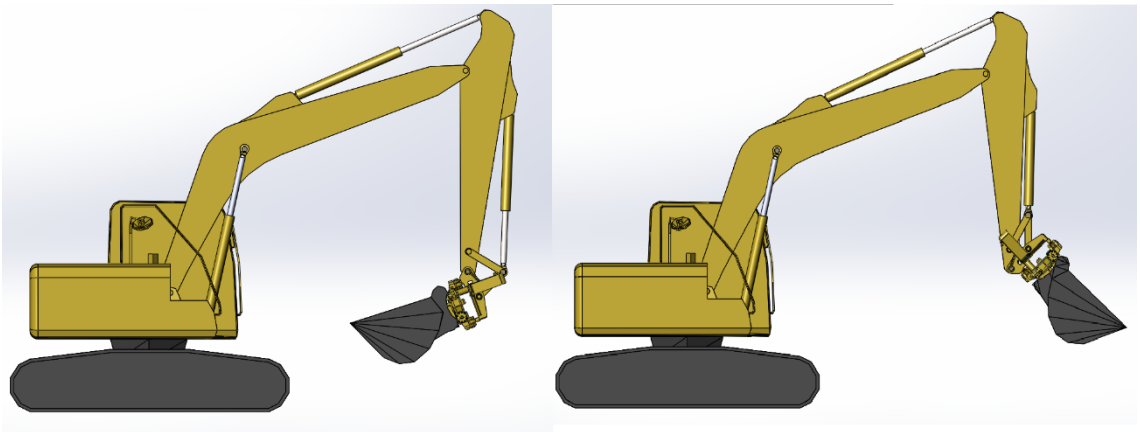

Figure 13. Excavator initial and final pose in bucket turn sequence

During the bucket turn sequence, the change in the joint angles is very straightforward leading to smooth transitions in the joint space. However, the corresponding change in the operational space is slightly more complex. The sequence in the joint space is presented in figure 19 and in the operational space in figure 20.

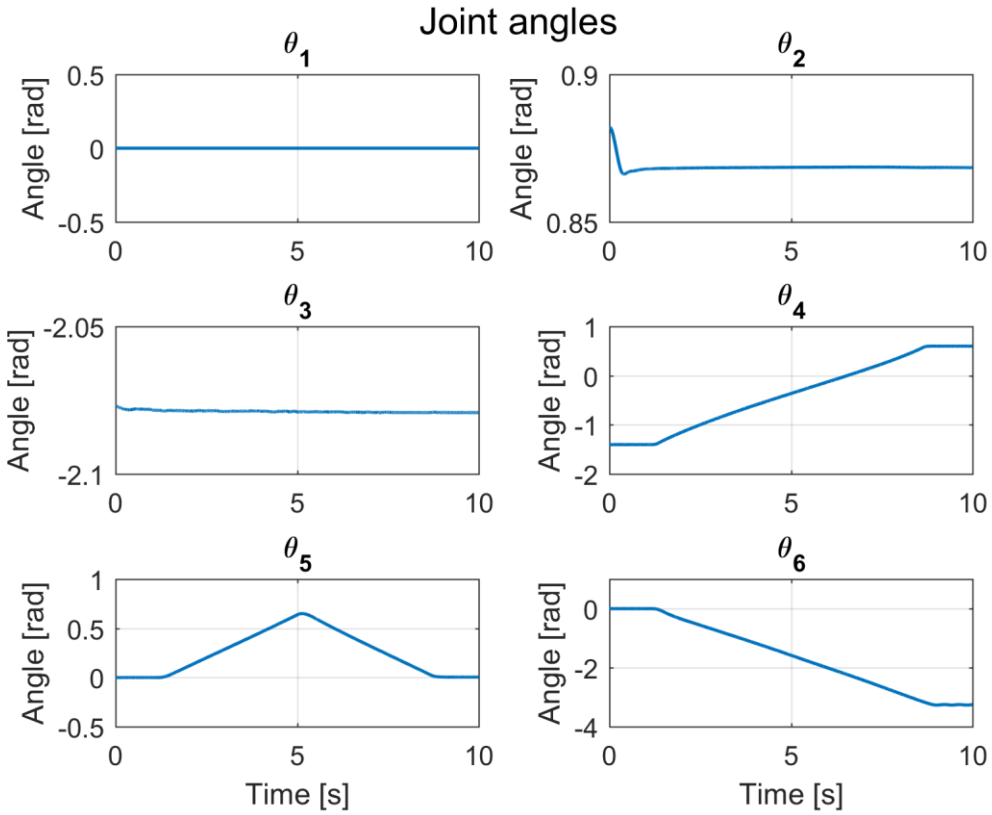

Figure 14. Bucket turn sequence in joint space

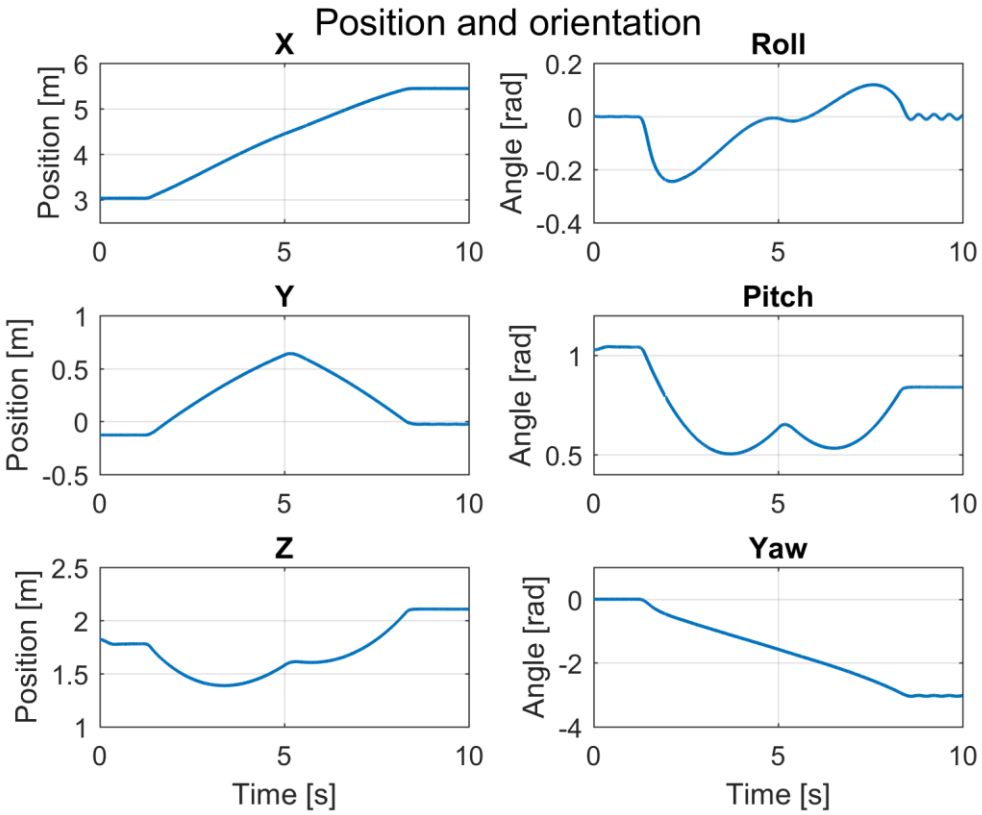

Figure 15. Bucket turn sequence in operational space

In figure 19, all the joint changes during the sequence are linear, making it very easy to construct a path in accordance. In figure 20, the change of the *z*-coordinate of the end-effector and the roll and pitch angles are functions of higher order, which makes it more difficult to construct a path for the end-effector to follow, or at least it would require a large number of intermediate points close together. Thus, from figures 19 and 20, it can be deducted that any possible trajectories involving the use of the bucket turn sequence are easier to construct and execute in the joint space rather than the operational space.

5.2 Square path

A very rudimentary method of testing a manipulator controller is to drive the manipulator on a square path in Cartesian space and compare the desired end-effector trajectory with the measured or in this case simulated trajectory. The reference for the orientation of the end-effector was chosen as constant for this simulation so that any changes in the end-effector pitch would directly result to the Cartesian trajectory paths for the *x*- and *z*-coordinates. The desired trajectory was given in the operational space and transformed to the joint space with the inverse kinematics after which it was fed as a reference to the PI-controller. The end-effector *x*- and *z*-coordinates and pitch angle with respect to time are presented in figure 21.

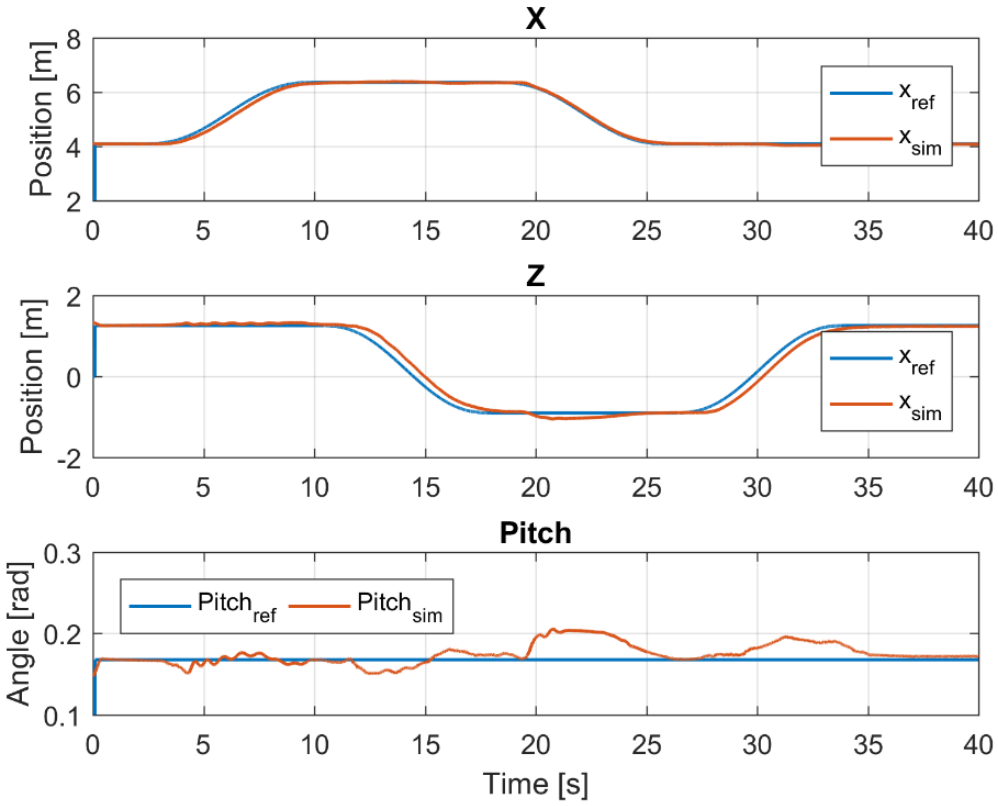

Figure 16. End-effector x- and z-coordinates and pitch angle on square path

In figure 21, both the simulated *x*- and *z*-coordinates lag their reference paths. This is caused by the combined effect of both the high mass and consequent inertia of the excavator links and the desired velocity reference of the end-effector trajectory. This lag could be attempted to eliminate by setting higher proportional and integrator gain values to the controller, but that will lead to unstable and undesired behavior of the excavator.

In addition to the lag in the transition phases, there is deviation present in the end-effector *z*-coordinate between the 20 and 25-second period. From figure 21, it can be seen that there is substantial deviation in the pitch angle of the end-effector during that time period, which directly causes error to the end-effector *z*-coordinate. The deviation between the reference trajectories and the simulated results are shown in figure 22.

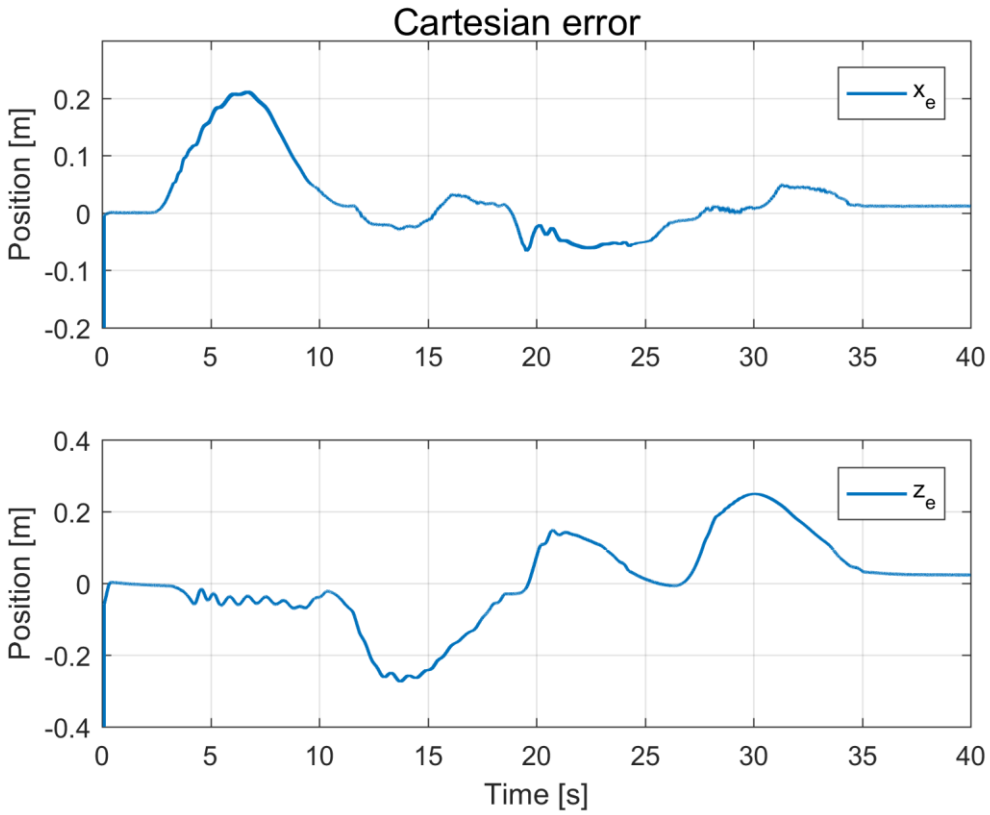

Figure 17. End-effector x- and z-coordinate error on square path

From figure 22, the end-effector *x*-coordinate follows the reference quite well excluding the transition from the first point to the second point of the path. However, the *z*-coordinate has substantial difficulties tracking the reference trajectory, especially when the movement is in the vertical plane, where the tracking error fluctuates between -0.27 and 0.25 meters.

5.3 Cartesian control

A convenient way to operate an excavator is to control the position of the end-effector along axes of a Cartesian coordinate frame with a single joystick direction command. In this control scheme, a velocity reference is given with the joystick, which is then integrated into a position reference and fed through the inverse kinematics to the PI-controller as a joint space reference trajectory.

When using a Cartesian control scheme, certain restrictions must be taken into account. Firstly, the joint velocity limitations of the manipulator have to be considered and adjust the maximum velocity reference from the joystick accordingly. Secondly, if the admissible workspace of the manipulator is not sufficiently defined, the Cartesian control scheme might attempt to drive the end-effector to a point which is outside the workspace or which leads to a kinematic singularity. In such a situation, the inverse kinematics may choke

and give an undesirable reference value to the PI-controller causing the manipulator to perform a rapid and radical movement.

The simulation model of the Komatsu excavator was simulated using a Cartesian control scheme. The trajectories of the end-effector *x*- and *z*-coordinate are presented in figure 23 and the resulting joint errors in figure 24.

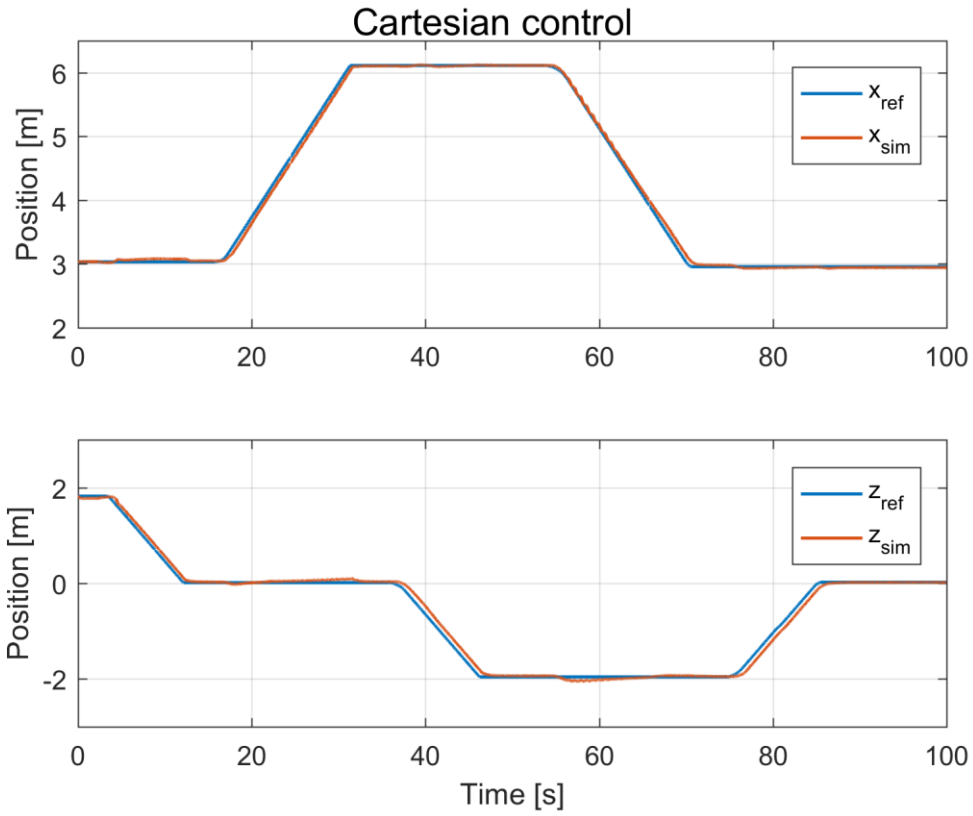

Figure 18. End-effector trajectories with a Cartesian control scheme

From figure 23, both the end-effector *x*- and *z*-coordinate follow the reference well. However, in the *z*-coordinate there is similar lagging visible as was with the square path simulations in section 5.2. Noteworthy of the simulated paths in figure 23, is that when moving along the *x*-axis, the *z*-coordinate stays constant and similarly when moving along the *z*-axis the *x*-coordinate stays constant. This relationship between the Cartesian coordinates is the basis of the Cartesian control scheme and based on the simulation, the control scheme is working well.

In figure 24, errors of joints θ_1, θ_5 and θ_6 were omitted from the simulation, as the angles were constant throughout the simulation. Observing the time period between 17 and 31 seconds in figure 23, the simulated *x*-coordinate has the same rise angle as the reference, but they are separated by a small bias. Observing the same time period in figure 24 reveals a constant joint error in joint θ_4 and a relatively constant error in joint θ_3. A similar pattern

appears also around the 40-second mark, where the *z*-coordinate of the end-effector changes over time.

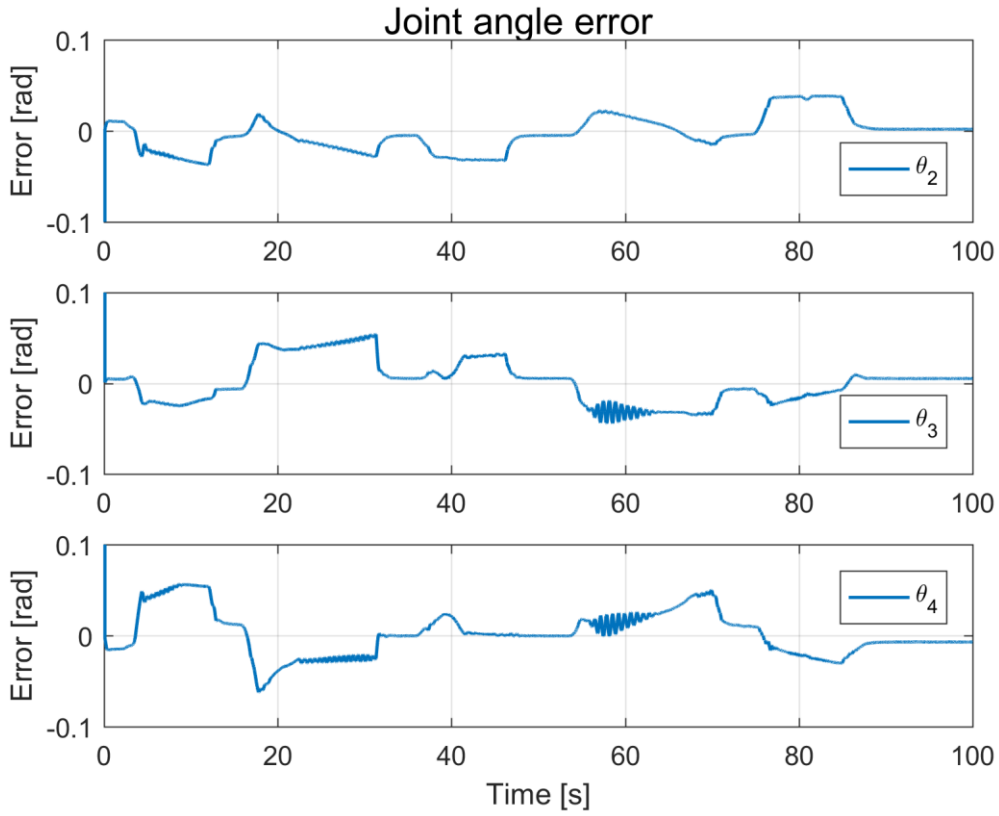

Figure 19. Excavator joint error with a Cartesian control scheme

At 54 seconds, small vibration appears in joint θ_2 and notable vibration in joints θ_3 and θ_4, which is clearly translated to the Cartesian paths of the end-effector. A key factor to this vibration is the configuration and direction of movement of the excavator at that point. At the beginning of the vibration, the excavator bucket is extended to over six meters and is two meters below the ground plane. This configuration combined with a movement in the negative *x*-axis direction forces the actuators to do positive work opposed to simply lowering the bucket in the negative *z*-direction, where gravity does the work.

5.4 Excavator use cases

The performance and accuracy of the excavator simulation model was tested with two characteristic use cases specifically designed to highlight and emphasize the nature of the Komatsu excavator studied in this thesis. Both use cases represent a typical setting an excavator might encounter on a daily basis for example at a construction site. The first use case utilizes the excavator inverse kinematics equations and the PI-controller, as well as the trajectory generator. Contrary to that, the second use case relies on the hybrid position/force controller introduced in section 4.2.

5.4.1 Dig-dump sequence

The first use case emulates a cycle, where gravel is collected in the bucket after which the bucket is turned around using the bucket turn sequence introduced in section 5.1., keeping the gravel in the bucket. The gravel is then dumped into the foundation of a building and the excavator is returned to the initial pose.

The digging and dumping sequences are created in the Cartesian space with the trajectory generator using intermediate path points, but the bucket turning sequence is done in the joint space due to the simpler nature of trajectory creation explained in section 5.1. The different stages of the full sequence are displayed in figures 25, 26, 27 and 28.

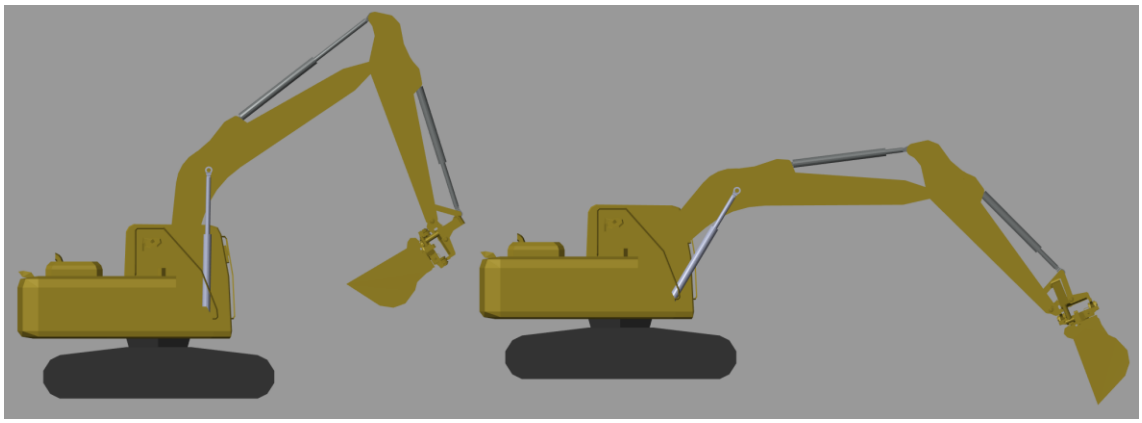

Figure 20. Initial pose and start of digging

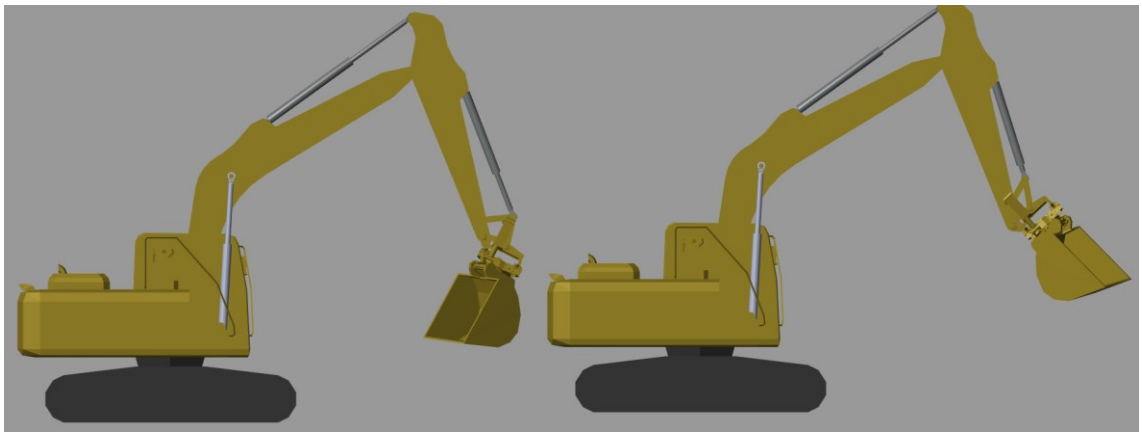

Figure 21. Bucket turn sequence

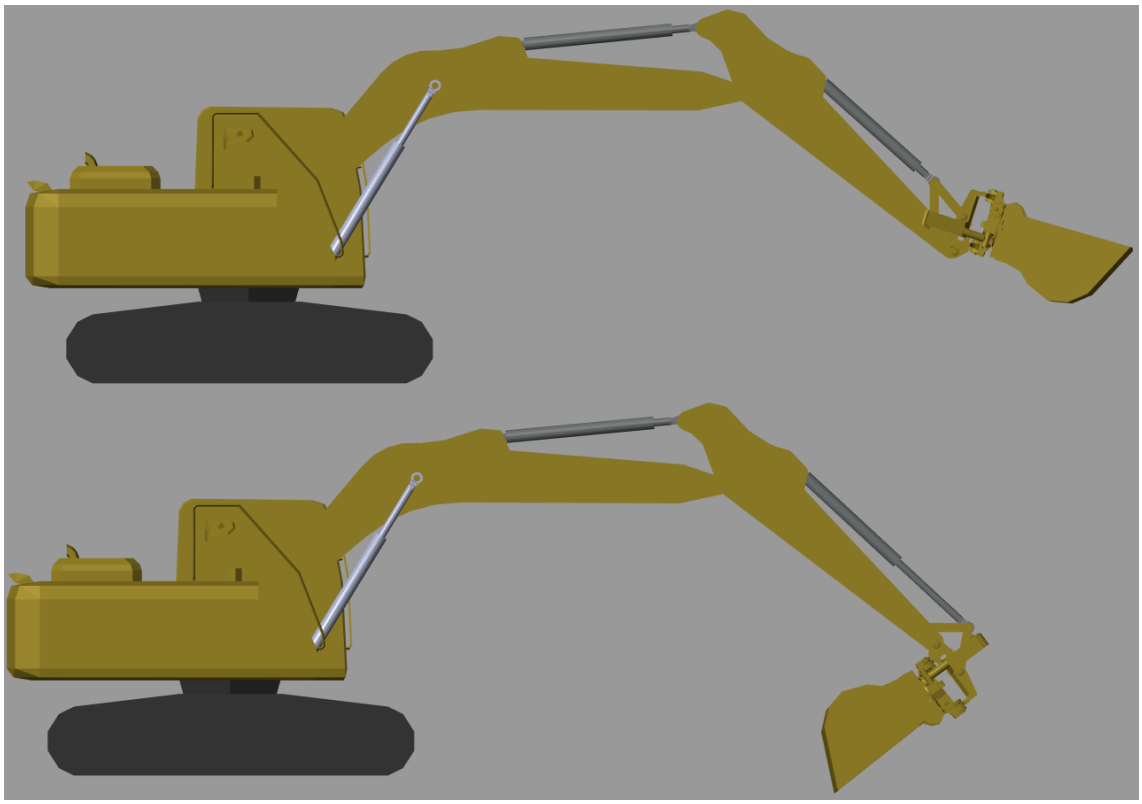

Figure 22. Extension to building foundation and dumping

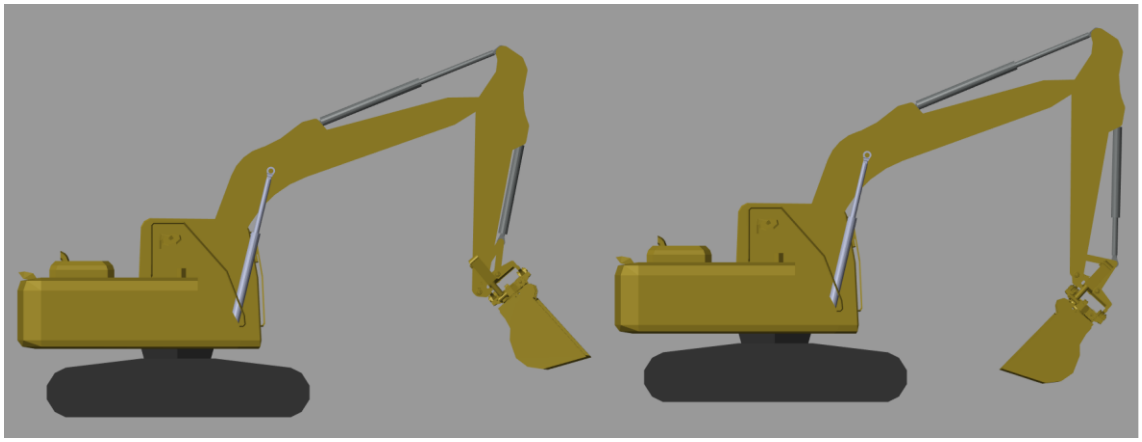

Figure 23. Returning to excavator initial pose

The full trajectory depicted in figure 29 was constructed in the joint space as a combination of the individual components of the sequence. The dig-dump sequence was conducted in the *xz*-plane, resulting in the cabin slew, joint variable θ_1, to be constant. Comparing the end-effector path in joint space shown in figure 29 and the path in the Cartesian space in figure 30 exposes the transitions between the different phases of the full sequence are smoother when executed in the joint space. For example at the 40 second point, right before the dumping phase the tiltrotator bucket rotation, joint variable θ_6, is constant but the end-effector roll angle has significant vibration and the yaw angle has slight vibration also.

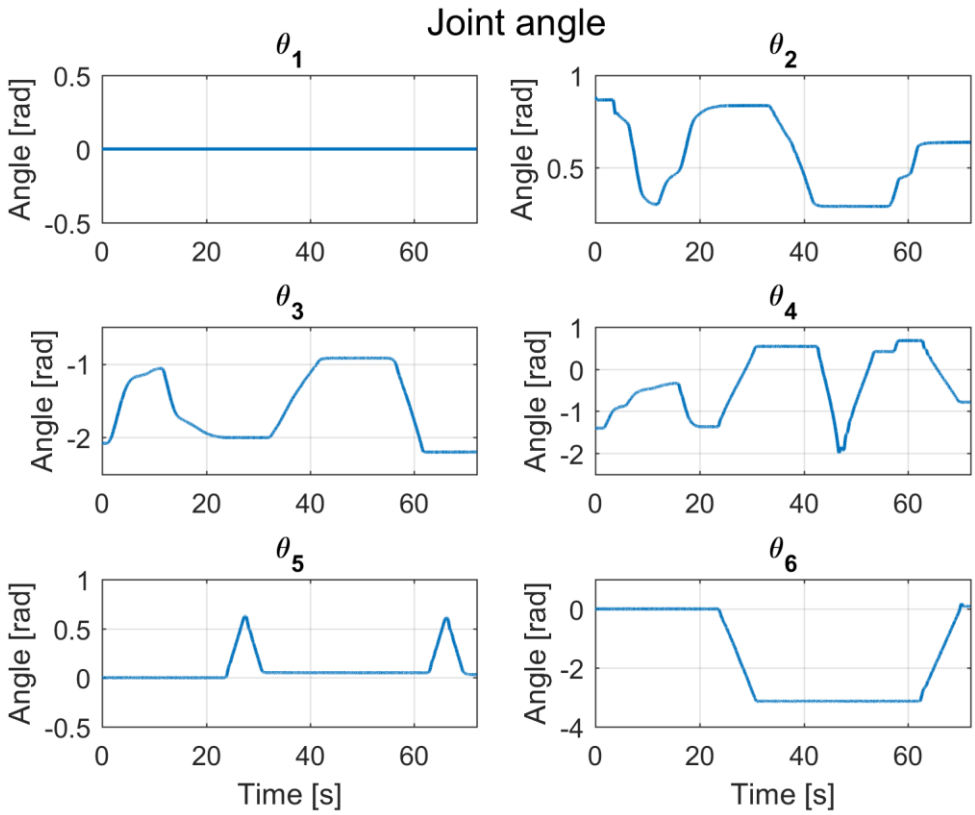

Figure 24. Dig-dump sequence trajectory in joint space

Notable similarities between the joint space path and the operational space path can be observed from figures 29 and 30. Since the cabin slew was restricted, the change in the *y*-direction in figure 30 is directly related to the tiltrotator tilt, joint variable θ_5, in figure 29. Moreover, the bucket rotation, joint variable θ_6, correlates to the end-effector yaw angle in figure 30 and the rapid change in the pitch angle in the dumping phase around the 40 second point in the path can be observed in the bucket joint, joint variable θ_4, in figure 29. The form of the end-effector *x*-coordinate path in the operational space resembles the path of joint variable θ_3 in the operational space as well as the form of the *z*-coordinate resembles the path of joint variable θ_2. However, direct correlation cannot be deducted since both the *x*- and *z*-coordinate are dependent on the combined result of joint variable θ_2, θ_3 and θ_4.

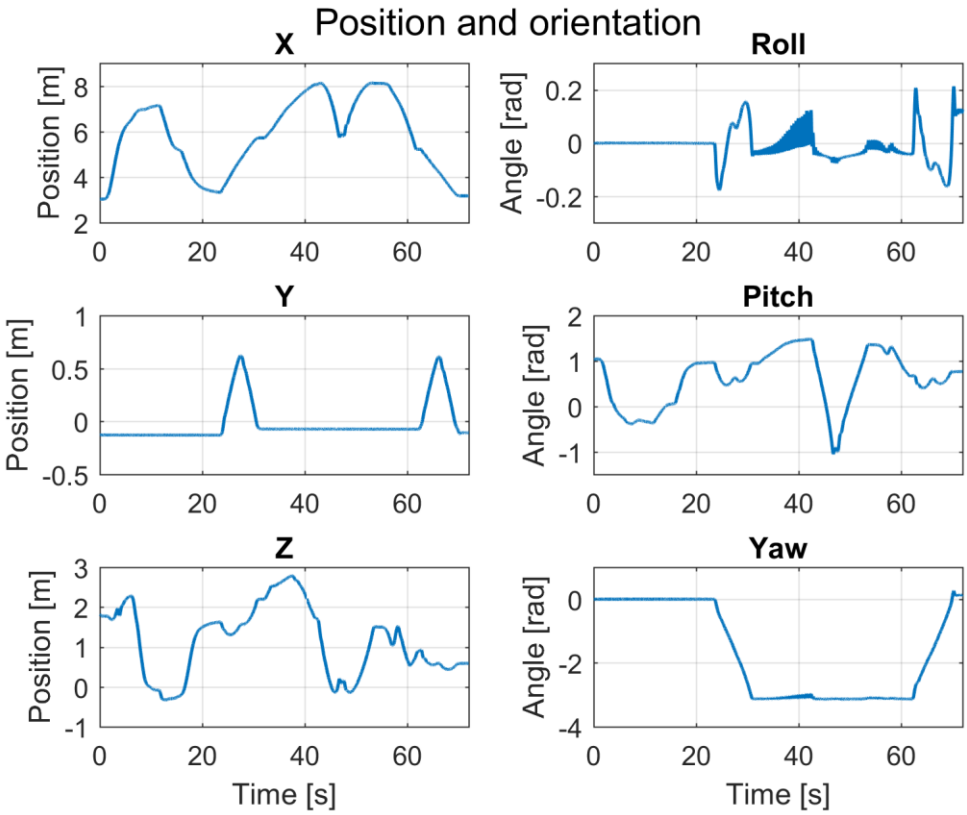

Figure 25. Dig-dump sequence position and orientation

The overall error in the joint space during the full dig-dump sequence is depicted in figure 31. During the excavator movement, the constant non-zero joint angle error sections in all joint variables reveal, that the high mass and consequent inertia of the excavator causes the end-effector to trail the reference trajectory. However, immediately after the movement ceases, the PI-controller rapidly adjusts the angles to the correct values. Since the controller does not have a derivative block implemented to it, steady-state error is present in all joint variables during phases where no movement is required of the joint in question.

The peak joint angle error during the entire dig-dump sequence is less than 0.1 radians, with the exception of joint variable θ_6, in which the joint angle error during the bucket turn sequence is 0.16 radians. Fine-tuning the controller proportional and integrator gains will result in better tracking accuracy, but the underlying cause of the tracking error is more likely the inaccuracy in the hydraulic valve and actuator parameters in the simulation model.

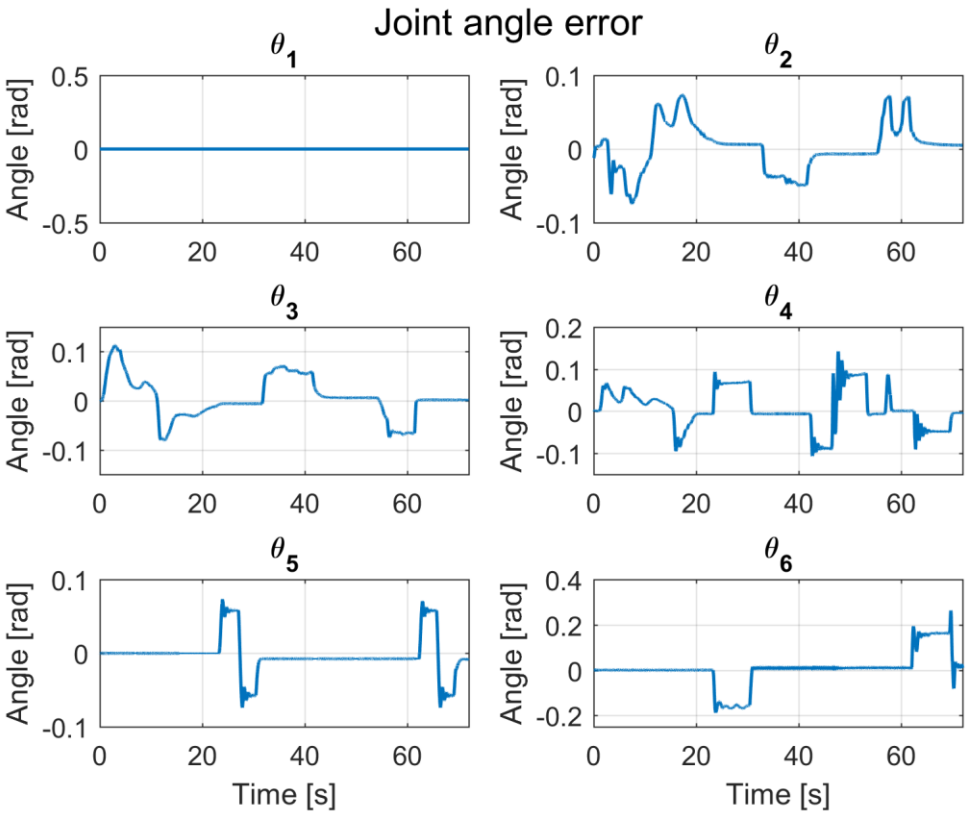

Figure 26. Joint angle error in dig-dump sequence

5.4.2 Digging sequence with contact force

The second use case in this thesis relies on the use of the hybrid position/force controller created in section 4.2. The key difference of the hybrid controller compared to the PI-controller used in the first use case is that the *z*-direction in the Cartesian space is now force controller and only the *x*- and *y*-direction are position controlled.

In the digging sequence the cabin slew, joint θ_1, as well as tiltrotator joints θ_5 and θ_6 are constant and the controller has been constructed as a four degrees of freedom construction. In the initial pose of the excavator, the bucket is extended to 7.6 meters in the horizontal *x*-direction and 0.1 meters in the vertical *z*-direction.

The end-effector trajectory along the *x*-axis is presented in figure 32. The radical movement in the beginning of the simulation is caused by the difference between the joint torques measured from the hydraulic cylinder chamber pressure differences and the torques obtained using the manipulator equations of motion. However, the controller obtains a steady state after which it can begin to track the desired position trajectory and force requirement.

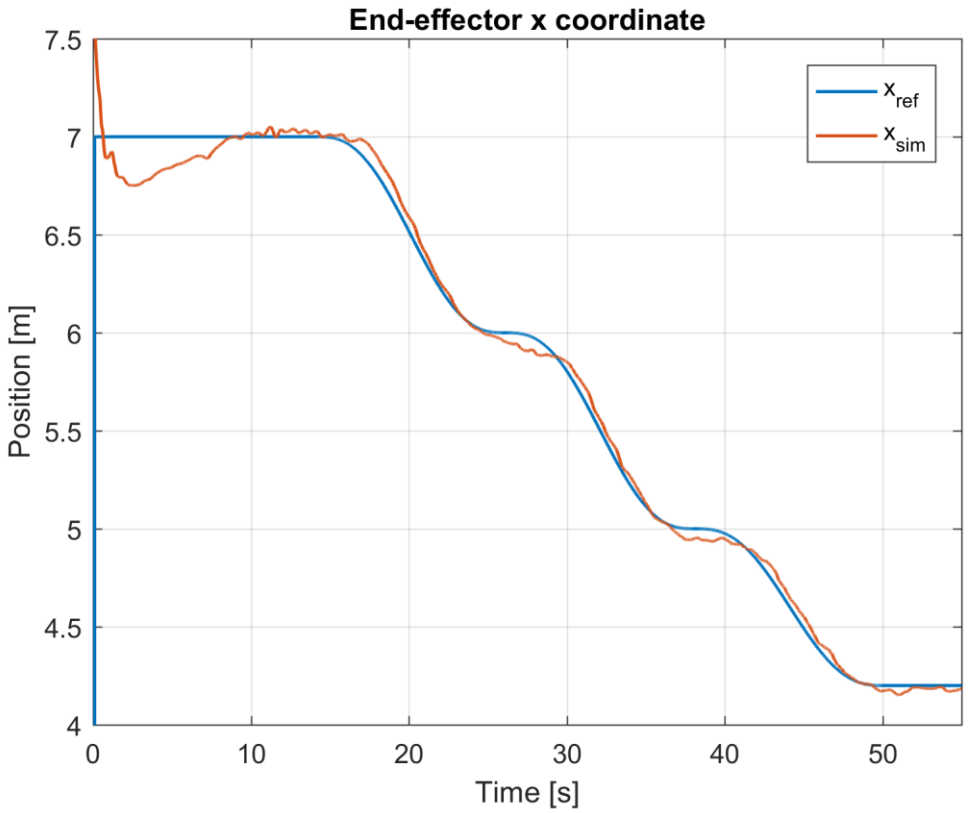

Figure 27. End-effector x-coordinate with hybrid position/force controller

In figure 32, excluding the obvious error in the beginning of the simulation, the controller tracks relatively well the desired trajectory. However, when there is a plateau in the desired path, the controller creates undesired error to the trajectory. Throughout the entire path and especially at sections where the *x*-coordinate reference is constant, noticeable vibration is present in the simulated path of the end-effector. The resulting joint angle errors are depicted in figure 33.

From figure 33, it can be observed that the upper position control loop works well as the maximum joint angle error during the work cycle is 0.025 radians in joint θ_3. The larger error during the beginning of the simulation is neglected as the reference trajectory is constant and the controller is still obtaining a steady state condition. The cumulative effect of the individual joint angle errors is better visualized in the Cartesian space. The end-effector error in the Cartesian space is presented in figure 34.

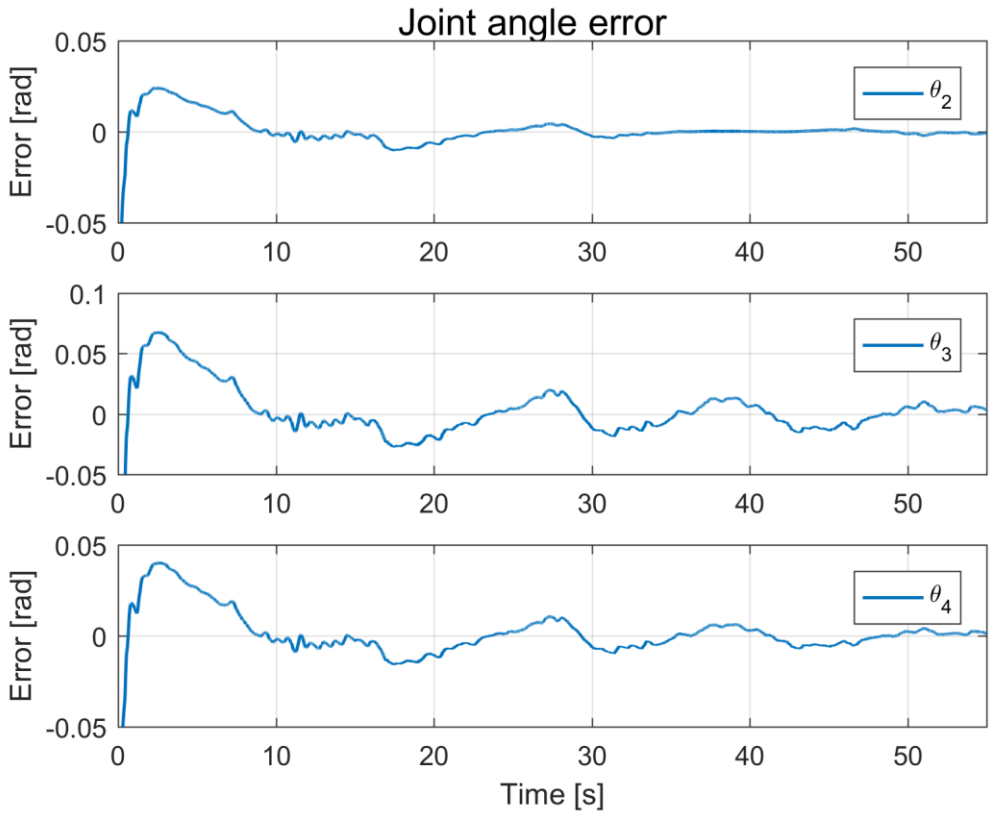

Figure 28. Joint angle error with hybrid position/force controller

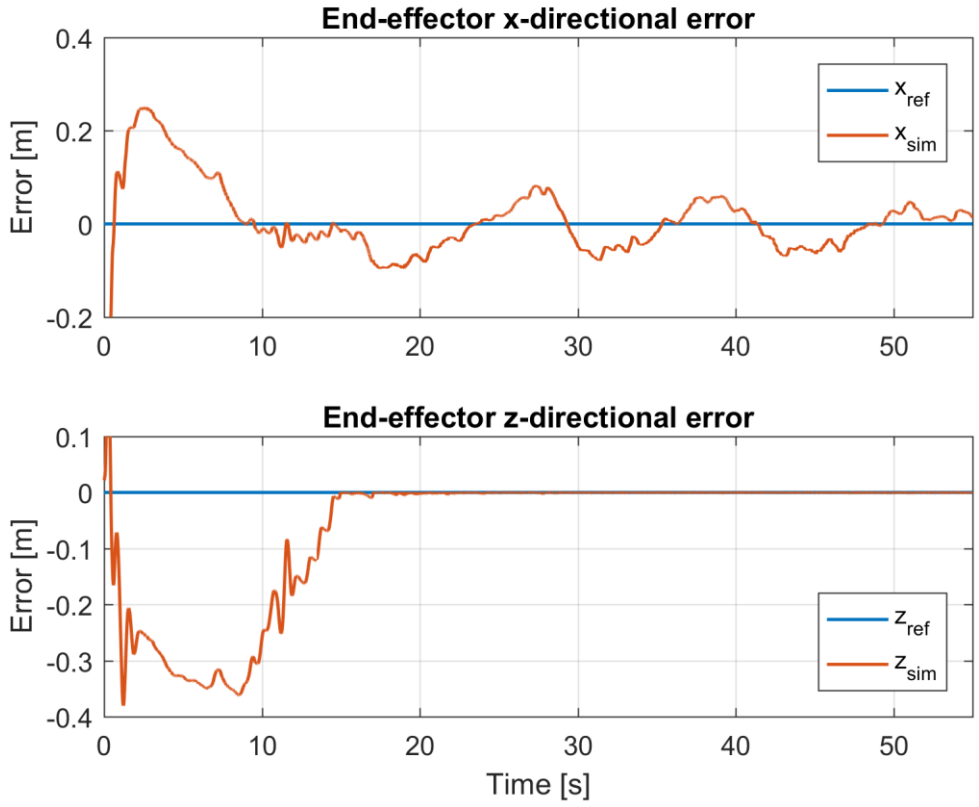

Figure 29. End-effector error in the Cartesian space

The maximum *x*-directional error during the digging cycle is 0.095 meters at the 17 second point. The *z*-coordinate builds substantial error during the beginning of the simulation, but sets at zero when the desired trajectory begins to take effect and as the end-effector makes contact with the ground.

The external force caused by digging, which was introduced in section 3.4., was implemented to the simulation model and fed to the lower force control loop as a force reference. The generated external force and the counter-acting controller force are presented in figure 35.

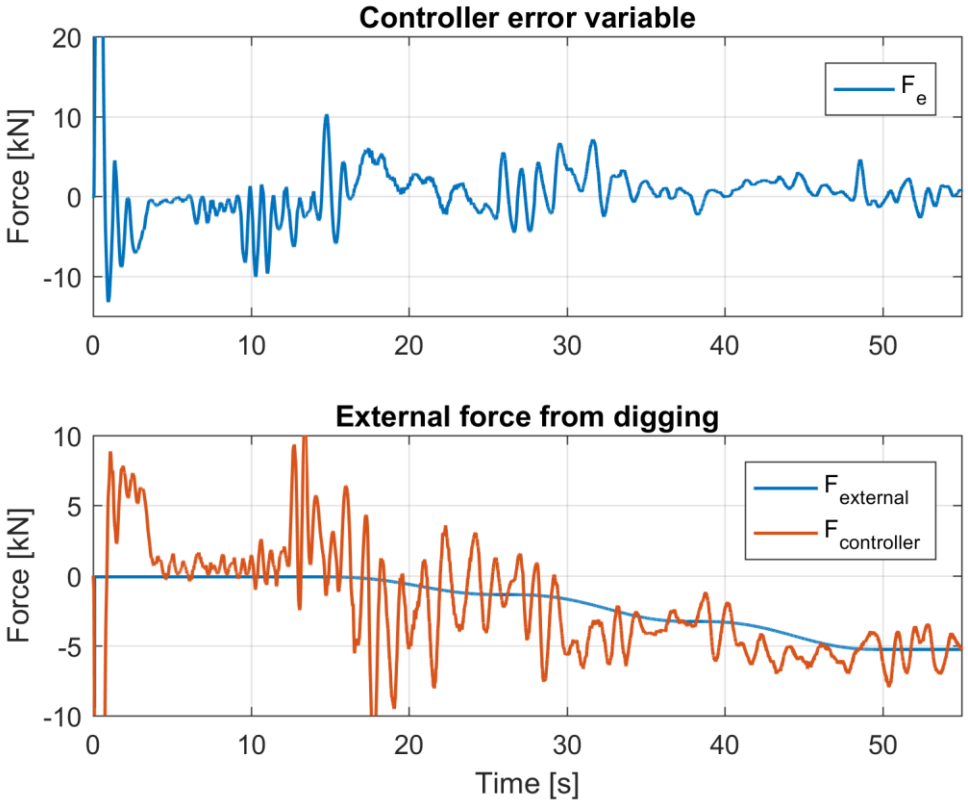

Figure 30. External reaction force from digging and lower force control loop force

In figure 35, the external reaction force caused by digging increases from zero to 5.3 kN at the end of the digging sequence. The balancing controller force follows the external reaction force trajectory, but it contains large vibration. A better implementation of the effects of the ground plane to the simulation model would reduce the amount of vibration as well as more precise tuning of the proportional and integral parameters, which were obtained experimentally in this case. The controller error variable F_e, which ideally should be zero, floats around zero with vibration caused by the aforementioned controller balancing force. Nonetheless, it can be observed that the division of end-effector degrees of freedom with the compliance selectivity matrix does work properly.

6. CONCLUSIONS AND FUTURE WORK

In this thesis, a simulation model for a Komatsu PC138US-8 excavator was constructed. To verify the performance and validity of the model, two closed-loop control schemes were created, and they were tested with use case scenarios in which an excavator might operate at a work site. The mechanical model of the excavator was assembled in Solid-Works CAD software and imported into MATLAB Simulink environment. The hydraulics of the excavator were constructed in Simulink using the excavator's hydraulics schematic as a reference.

First off, the forward and inverse kinematics were formulated. With these kinematical equations, the excavator's operational space and joint space can be linked together, providing a foundation for the controller design implemented in this thesis. While the forward kinematics equations work impeccably, the closed form inverse kinematics have some discrepancies to them. In static conditions the inverse kinematics yield the correct solution for the excavator joint angles, but under dynamic situations, especially when the tiltrotator bucket turn joint is applied, the solution may be outside the excavator's admissible working area. Another issue, caused by the definition of the Atan2-function, is the joint value range near $\pm\pi$. Crossing over either one causes the joint reference to change by 2π, leading to rapid and undesired control references to the excavator valves. These issues can be overcome with better definition of the joint angle restrictions.

In addition to the kinematics, the dynamics of the excavator are mandatory in order to construct the desired controller schemes in this thesis. The dynamics were derived using Lagrangian equations of motion using the mass and dimension properties obtained from the CAD model.

The first use case studied in this thesis was a digging and dumping sequence, which utilized the tiltrotator to turn the bucket from a backwards position to a forward position. The reference trajectory was constructed from four separate sections, which were combined to form the full reference trajectory. The digging and dumping sections of the full sequence were constructed in the Cartesian space using the inverse kinematics equations and the PI-controller, and the bucket turning sections were constructed in the joint space. The full trajectory was then constructed in the joint space. Based on the simulation results, the excavator follows the reference trajectory well. However, fast movement on the trajectory causes the joint angles to lag the reference momentarily leading to small error in the joint value. In addition, minute steady state error is present in the joint values due to the lack of a derivative block in the implemented controller.

The second use case studied relied on the use of a hybrid position/force controller in a digging operation, where contact with the ground inflicts external force on the excavator

end-effector. The x-directional position reference trajectory was given in the Cartesian space and the external contact force was given as the *z*-directional force reference. The simulation results show that the position tracking was good as the upper level position control loop kept the joint error under 0.025 radians during the digging operation. The lower level force control loop followed the force reference trajectory well, but quite large vibration was present in the controller force signal.

In the future, some improvements should be made to the simulation model constructed in this thesis. The mass and inertia properties no not match the actual excavator perfectly and the implemented hydraulics are still insufficient. Furthermore, the minor shortcomings of the inverse kinematics need to be addressed and the hybrid controller extended from the current 4DOF to a 6DOF construction. Taking measurements from the actual excavator will help increase the accuracy of the simulation model and eventually lead to the possible implementation of the control schemes to the actual excavator.

REFERENCES

[1] CAT Excavators. Equipment. 2017. Available: http://www.cat.com/en_US/products/new/equipment/excavators.html. Accessed: 11.7.2017

[2] L. Sciavicco and B. Siciliano, Modeling and Control of Robot Manipulators, 2nd Edition, Springer, 378p

[3] S. Baglioni, F. Cianetti, C. Braccesi, L. Landi, Parametric Multibody Modeling of Antropomorphic Robot to Predict Joint Compliance Influence on End Effector Positioning, ASME International Mechanical Engineering Congress & Exposition, November 2013

[4] J. Koivo, Fundamentals for Control of Robotic Manipulators, Wiley, 468p

[5] R. Jazar, Theory of Applied Robotics, Springer, 883p

[6] J. Kuipers, Quaternions and Rotation Sequences, Department of Mathematics, Calvin College, Grand Rapids, MI, September 1999. Available: http://www.emis.ams.org/proceedings/Varna/vol1/GEOM09.pdf. Accessed: 18.7.2017

[7] S. Trafazoli, P. Lawrence, S.E. Salcudean, Identification of Inertial and Friction Parameters for Excavator Arms, IEEE Transactions on Robotics and Automation, Vol 15, no. 5, October 1999

[8] H. Gavin, The Principle of Virtual Work, Duke University, Department of Civil and Environmental Engineering, Matrix Structural Analysis, 2012. Available: http://people.duke.edu/~hpgavin/cee421/virtual-work.pdf. Accessed: 2.8.2017

[9] J. J. Craig, Introduction to Robotics, 2nd edition, Addison-Wesley Publishing Company Inc., 450p

[10] J. J. Craig, M. H. Raibert, Hybrid Position/Force Control of Manipulators, Journal of Dynamic Systems, Measurement, and Control, Vol. 102, June 1981

[11] M. Vukubratovic, A. Tuneski, Contact Control Concepts in Manipulator Robotics – An Overview, IEEE Transactions on Industrial Electronics, Vol. 41, February 1994

[12] J. E. Jam, A. A. Fard, Application of Single Unit Impact Dampers to Reduce Undesired Vibration of the 3R Robot Arms, International Journal of Aerospace Sciences, February 2013

[13] A. J. Koivo, M. Thoma, E. Kocaoglan, J. Andrade-Cetto, Modelin and Control of Excavator Dynamics During Digging Operation, Journal of Aerospace Engineering, January 1996

[14] J. Blanco, A Tutorial on SE(3) transformation parameterizations and on-manifold optimization, University of Malaga, Department of Systems Engineering and Automation, October 2014. Available: https://pixhawk.org/_media/dev/know-how/jlblanco2010geometry3d_techrep.pdf. Accessed: 27.7.2017

[15] R. Murray, Control and Dynamical Systems, Robot Dynamics and Control. Available: http://www.cds.caltech.edu/~murray/books/MLS/pdf/mls94-manipdyn_v1_2.pdf. Accessed: 13.7.2017

[16] Komatsu PC138US-8 Datasheet. Available: http://www.komatsu.com/ce/products/pdfs/PC138US-8_CEN00183-06.pdf. Accessed: 15.6.2017

APPENDIX A: SIX DEGREES OF FREEDOM TRAJECTORY GENERATOR

The trajectory generator utilized in this thesis is based on a five degree quintic polynomial path. Because the degree of the polynomial path is five, the number of required conditions to satisfy it is n+1=6. These six conditions come from specified position, velocity and acceleration values at set boundaries, which can be expressed as:

$$q(t_0) = q_0 \qquad \dot{q}(t_0) = q_{0\_vel} \qquad \ddot{q}(t_0) = q_{0\_acc}$$
$$q(t_f) = q_f \qquad \dot{q}(t_f) = q_{f\_vel} \qquad \ddot{q}(t_f) = q_{f\_acc} \tag{A.1}$$

where t_0 is the initial position and t_f the final position. [5]

Now, a five degree polynomial to satisfy the conditions in equation (A.1) can be constructed as:

$$q(t) = a_0 + a_1 t + a_2 t^2 + a_3 t^3 + a_4 t^4 + a_5 t^5 \tag{A.2}$$

where t is the time and a_i the quintic path coefficients.

Observing equation (A.1) it can be seen that by determining initial and final values for the position, velocity and acceleration, a path can be constructed between these two points with equation (A.2). Combining multiple of these intermediate points leads to the definition of the complete trajectory. According to [5] the quintic path coefficients determining the behavior of the path can be solved with the set of six equations as:

$$\begin{bmatrix} 1 & t_0 & t_0^2 & t_0^3 & t_0^4 & t_0^5 \\ 0 & 1 & 2t_0 & 3t_0^2 & 4t_0^3 & 5t_0^4 \\ 0 & 0 & 2 & 6t_0 & 12t_0^2 & 20t_0^3 \\ 1 & t_f & t_f^2 & t_f^3 & t_f^4 & t_f^5 \\ 0 & 1 & 2t_f & 3t_f^2 & 4t_f^3 & 5t_f^4 \\ 0 & 0 & 2 & 6t_f & 12t_f^2 & 20t_f^3 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \\ a_4 \\ a_5 \end{bmatrix} = \begin{bmatrix} q_0 \\ q_{0\_vel} \\ q_{0\_acc} \\ q_f \\ q_{f\_vel} \\ q_{f\_acc} \end{bmatrix} \tag{A.3}$$

For example, a rest-to-rest path with zero velocity and acceleration at the rest positions solved with equation (A.3) and satisfying the conditions:

$$q(0) = -2 \qquad \dot{q}(0) = 0 \qquad \ddot{q}(0) = 0$$
$$q(1) = 9 \qquad \dot{q}(1) = 0 \qquad \ddot{q}(1) = 0 \tag{A.4}$$

results in:

$$\begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \\ a_4 \\ a_5 \end{bmatrix} = \begin{bmatrix} -2 \\ 0 \\ 0 \\ 110 \\ -165 \\ 66 \end{bmatrix} \tag{A.5}$$

which yields the polynomial equation for the path as:

$$q(t) = -2 + 110t^3 - 165t^4 + 66t^5 \tag{A.6}$$

The path in equation (A.6) and its first and second derivative are shown in figure A1.

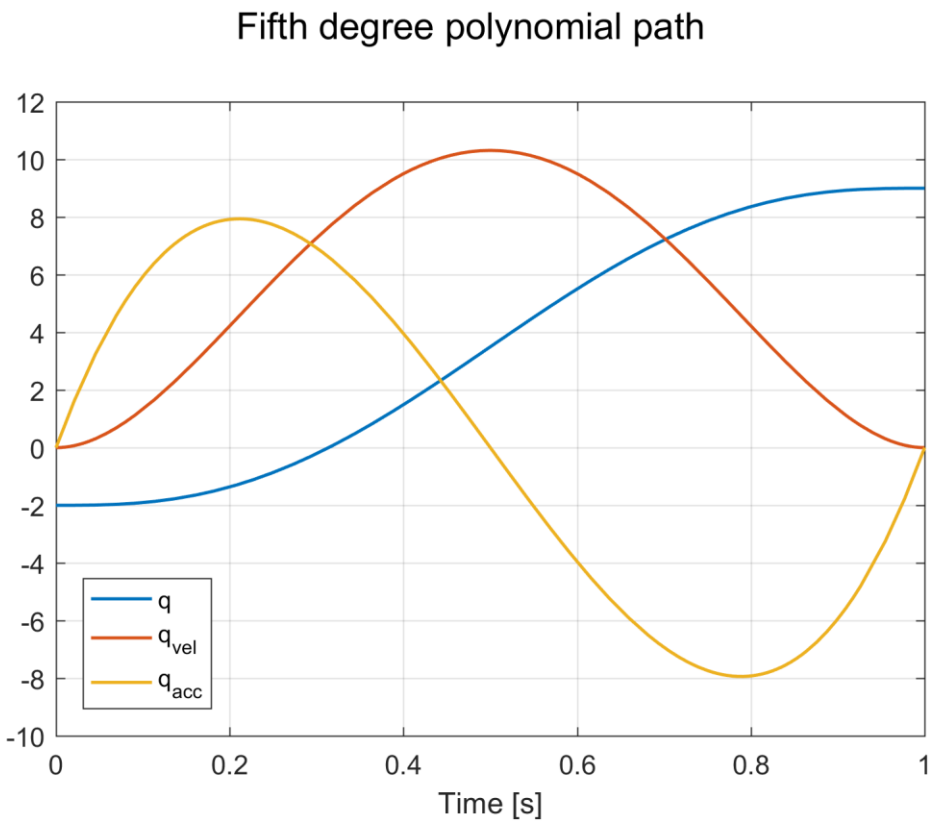

Figure A1. Position, velocity and acceleration of a fifth degree polynomial path

APPENDIX B: EXCAVATOR CAD MODEL

The Komatsu PC138US-8 excavator studied in the thesis was modeled and constructed with SolidWorks 2016 CAD software and imported into MATLAB Simulink as stereolithography files (.stl file extension) for the visualization and a separate data file for the information about the excavator configuration. At the time of writing this, no complete CAD model of the exact excavator was available, so a CAD model of a Komatsu PC160 excavator was used as a starting point instead. The excavator CAD model is presented in figure B1.

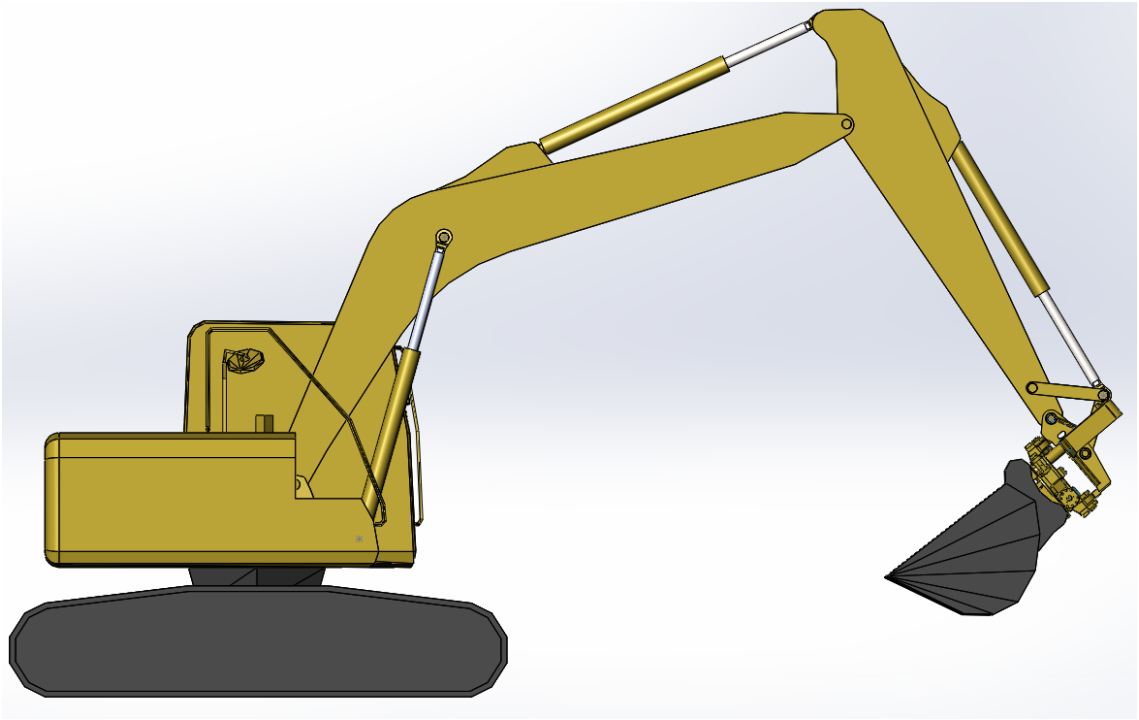

Figure B1. Excavator CAD model

From the PC160 model, the undercarriage, cabin and bucket were taken and scaled down to correct size. However, for the boom and stick, only the shape of the parts was taken as a reference and the parts were then entirely recreated from scratch. In addition, all hydraulic cylinders of the excavator were constructed from scratch. The PC160 excavator model was not equipped with a tiltrotator, but a CAD model was available for the specific model and it was added to the final assembly.

Once all the parts of the excavator were created, they were bound together with constraints called mates. With the mates, the revolute joints of the excavator, which also are the joint variables of the simulation model, can be established and defined in space. In addition, the hydraulic cylinders and pistons were constrained together with prismatic joint mates. In total, 52 mates were used to define the configuration of the excavator.

APPENDIX C: EXCAVATOR MECHANICAL MODEL IN MATLAB SIMULINK

The excavator model constructed in SolidWorks as a CAD assembly was imported into MATLAB Simulink using the Simscape Multibody Link plugin. Upon import, rigid coordinate transformation frames containing information of the translation and rotation between the two adjacent frames connect the .stl files containing the visual properties of each individual component of the assembly. The mass and inertia properties of each assembly element are contained in a parameter file along with the initial configuration states for each joint in the assembly.

In the import process, the type of joints resulting in the simulation model in Simulink is dependent on the mates created in the CAD software. In the case of the excavator, only pure revolute or pure prismatic joints were desired to be in the assembly, which led to very detailed mate definition. For example, if the position of a revolute joint is not fully defined along the rotational axis in the CAD software, the joint will be imported as a cylindrical joint with two degrees of freedom instead of a revolute joint with only one degree of freedom. This will lead to the simulation model to be unstable and possibly crash, because kinematic constraints cannot be maintained properly.

With the Simmechanics joint blocks, it is possible to directly get the position, velocity, acceleration and torque or force, depending on the type of joint in question, as an output variable. Moreover, for example a prismatic joint can be directly manipulated either with an applied force or with a motion input command, making it very versatile to use. The joint position and velocity information of the prismatic joints describing the hydraulic cylinders and the revolute joints describing the hydraulic motors are fed as a feedback signals to the hydraulics subsystem where the resulting forces based on the position and velocity information are calculated and fed back to the mechanical model as force and torque input signals.

The configuration of the excavator mechanical model created in Simulink is presented in figure C1.

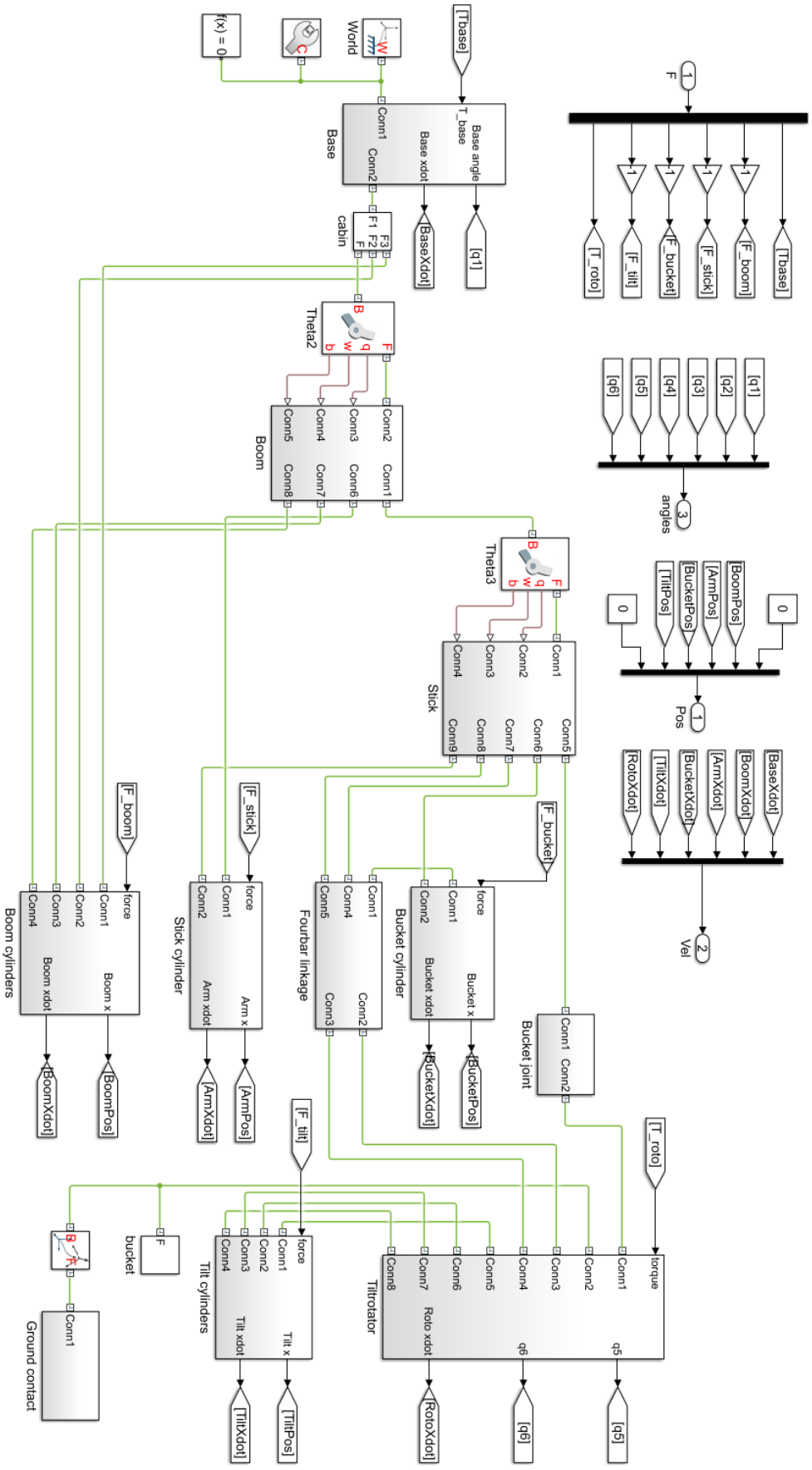

Figure C1. Excavator mechanical model in MATLAB Simulink

APPENDIX D: MODEL HYDRAULICS IN MATLAB SIMULINK

The hydraulics used in the simulation model in this thesis were created using the basic Simulink library blocks. All six hydraulic valves controlling the linear and rotational actuators were constructed as a 4/3 configuration, meaning that there are four ports and three positions in the valve. The shape of the valve stem and the resulting non-linear flow coefficient equations, namely the valve relative openings, were taken care of with separate look-up tables for each port connection combination. The configuration of the hydraulic valves is presented in figure D1.

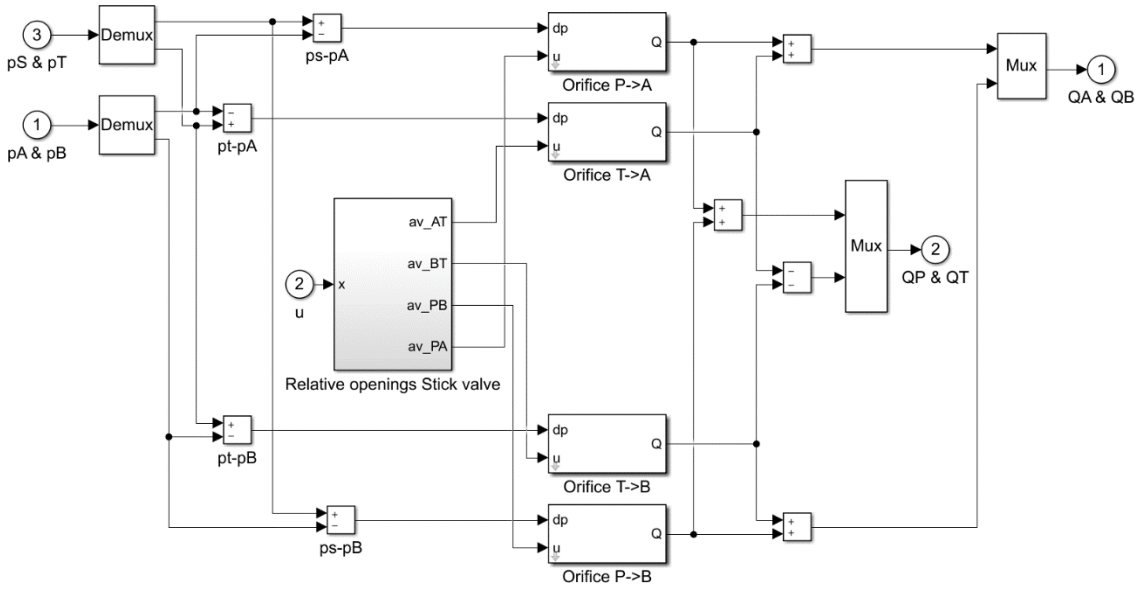

Figure D1. 4/3 hydraulic valve with look-up tables for relative valve openings

The dynamic behavior of the hydraulic valves was accounted for with second order transfer functions with a nominal frequency of 2 Hz.

The double acting hydraulic cylinders were modeled as a volume, whose pressure in both chambers is created by the time dependent volumetric change caused by the change in the cylinder stroke commanded by the feedback signal from the excavator mechanical model. In addition to the individual cylinder chambers, a hyperbolic tangent function is added alongside to emulate the effects of static and dynamic friction inside the cylinder. Moreover, a spring-mass function describing the hard stops at both zero stroke and maximum stroke situations is added to the overall resulting cylinder force. The hydraulic cylinders used in the simulation model are presented in figure D2.

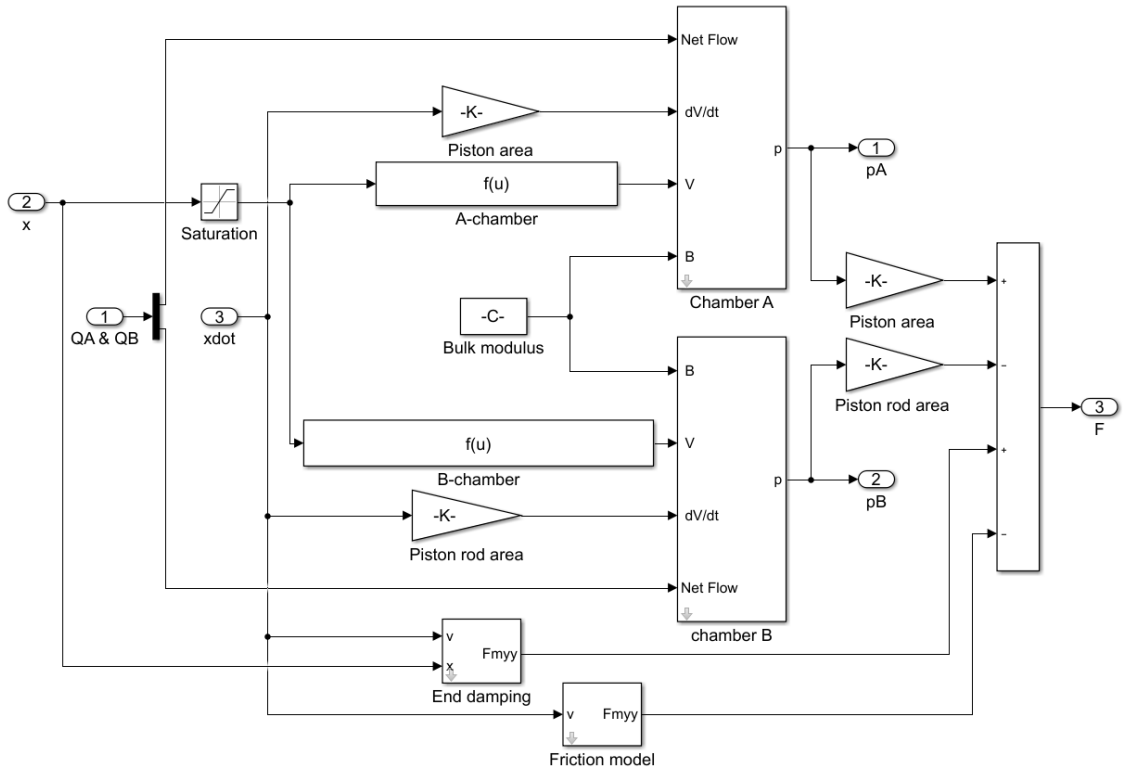

Figure D2. Double acting hydraulic cylinder with end damping and friction modeling

The resulting cylinder force, which is fed as a reference input to the mechanical model of the simulation model, is a sum of the individual chamber forces, friction and the end damping force.

The hydraulic motor model used in the tiltrotator bucket turn and cabin slew is presented in figure D3.

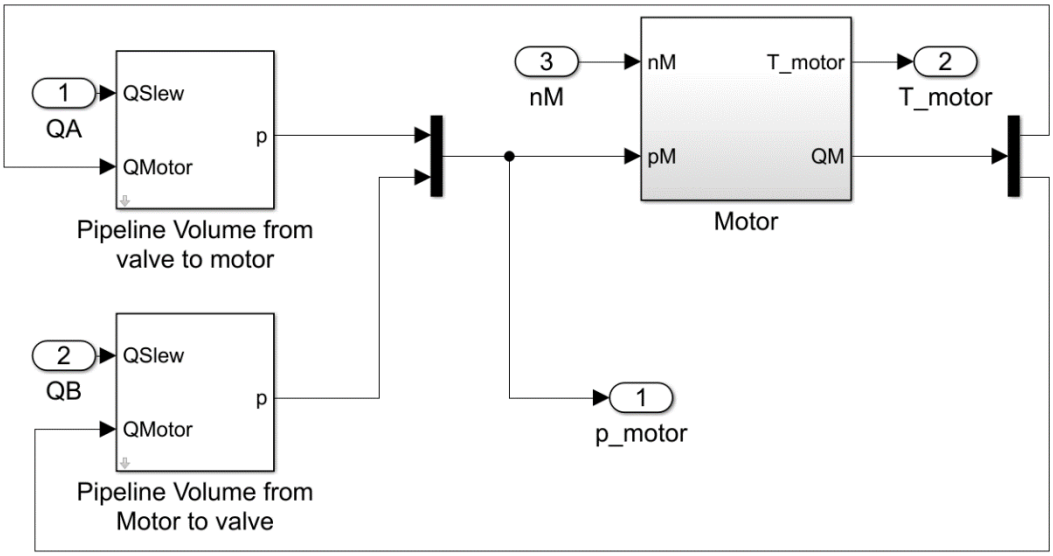

Figure D3. Hydraulic motor model in MATLAB Simulink

In a similar manner as with the double acting hydraulic cylinders, pressure is created using a volume and time dependent volumetric flow. The motor output volumetric flow can then be calculated using the pressure signal and the known motor volume size. With the motor volume size, also the output torque can be obtained by multiplying it with the nominal rotation speed, which is received from the mechanical model as feedback.

The overview of the hydraulics created in Simulink for the stick cylinder is presented in figure D4.

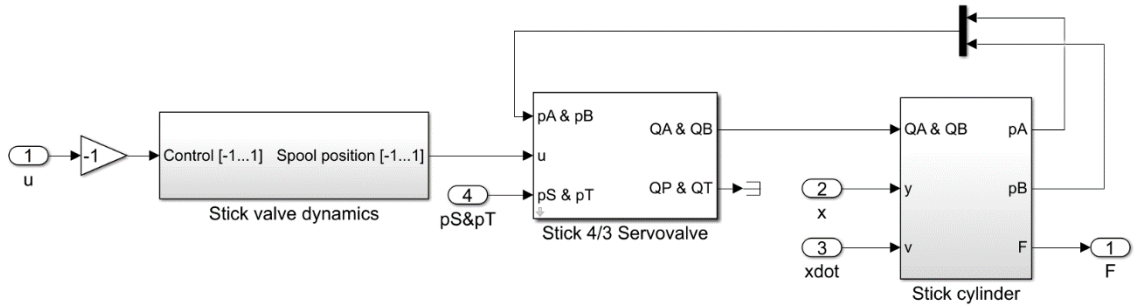

Figure D4. Overview of stick cylinder valve and actuator model from valve control input u to output force F

All remaining valve and actuator pairs are constructed in a similar manner as in figure D1.

APPENDIX E: TILTROTATOR BUCKET TURNING CONTROL SCHEME

The synchronous operation of the bucket joint, tiltrotator tilt and tiltrotator rotation enables the bucket to be turned around keeping the payload in the bucket. Joint variable θ_6, which is the bucket turn, controls the sequence the tilt operates in and also the direction and behavior of the bucket joint. The control scheme for the operation is presented below in this Appendix.

```
function u_control = command(bucket, joystick, tilt_angle, roto_angle)

%This function operates the last three joints of the excavator in a
%simultaneous and synchronous manner enabling the turning of the
%bucket from the back position to the forward position.
%Input "bucket" is a constant zero parameter
%Input "joystick" is the control value from the operator
%Inputs "tilt_angle" and "roto_angle" are measured joint values from
%the excavator
if bucket == 0
    if ((roto_angle > 0) && (roto_angle < pi)) || ((roto_angle < -pi)
&& (roto_angle > -2*pi))
        if joystick > 0
            u4 = 0.55*joystick;
        elseif joystick < 0
            u4 = 0.4*joystick;
        else
            u4 = 0;
        end
    elseif ((roto_angle < 0) && (roto_angle > -pi)) || ((roto_angle >
pi) && (roto_angle < 2*pi))
        if joystick > 0
            u4 = -0.4*joystick;
        elseif joystick < 0
            u4 = -0.55*joystick;
        else
            u4 = 0;
        end
    else
        u4 = 0;
    end
else
    if bucket > 0
        u4 = bucket;
    elseif bucket < 0
        u4 = bucket;
    else
        u4 = 0;
    end
end

if ((-pi/2 < roto_angle) && (roto_angle < pi/2)) || ((3*pi/2 <
roto_angle) && (roto_angle < 5*pi/2)) || ((-3*pi/2 > roto_angle) &&
(roto_angle > -5*pi/2))
        if joystick > 0
            if ((tilt_angle < 0.06) && (tilt_angle > -0.06)) && (((-
1.47 < roto_angle) && (roto_angle < -0.1)) || ((4.81 < roto_angle) &&
(roto_angle < 6.18)))
```

```
                u5 = 0;
                u6 = 0.45*joystick;
            else
                u5 = -0.4*joystick;
                u6 = 0.45*joystick;
            end
        elseif joystick < 0
            if ((tilt_angle < 0.06) && (tilt_angle > -0.06)) &&
(((1.52 > roto_angle) && (roto_angle > 0.1)) || ((-4.71 > roto_angle)
&& (roto_angle > -6.18)))
                u5 = 0;
                u6 = 0.45*joystick;
            else
                u5 = -0.4*joystick;
                u6 = 0.45*joystick;
            end
        else
            u5 = 0;
            u6 = 0;
        end
elseif ((pi/2 < roto_angle) && (roto_angle < 3*pi/2)) || ((-pi/2 >
roto_angle) && (roto_angle > -3*pi/2))
        if joystick > 0
            if ((tilt_angle < 0.06) && (tilt_angle > -0.06) && (((1.57
< roto_angle) && (roto_angle < 3.04)) || ((-4.71 < roto_angle) &&
(roto_angle < -3.04)))
                u5 = 0;
                u6 = 0.45*joystick;
            else
                u5 = 0.4*joystick;
                u6 = 0.45*joystick;
            end
        elseif joystick < 0
            if ((tilt_angle < 0.06) && (tilt_angle > -0.06)) &&
(((4.71 > roto_angle) && (roto_angle > 3.24)) || ((-1.51 > roto_angle)
&& (roto_angle > -3.04)))
                u5 = 0;
                u6 = 0.45*joystick;
            else
                u5 = 0.4*joystick;
                u6 = 0.45*joystick;
            end
        else
            u5 = 0;
            u6 = 0;
        end
else
    if joystick > 0
        u5 = 0;
        u6 = 0;
    elseif joystick < 0
        u5 = 0;
        u6 = 0;
    else
        u5 = 0;
        u6 = 0;
    end
end

u_control = [u4;u5;u6];
end
```